Collaborative Project

Holistic Benchmarking of Big Linked Data

Project Number: 688227 **Start Date of Project:** 2015/12/01 **Duration:** 36 months

Deliverable 3.2.2
Second Version of the Data Extraction Benchmark for Unstructured Data

Dissemination Level	Public
Due Date of Deliverable	Month 30, 29/05/2018
Actual Submission Date	Month 30, 29/05/2018
Work Package	WP3 - Benchmarking Platform
Task	T3.2
Type	Other
Approval Status	Final
Version	1.0
Number of Pages	23

Abstract: This deliverable presents the second version of the data extraction benchmark for unstructured data.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688227.

History

Version	Date	Reason	Revised by
0.1	02/05/2018	First draft created	René Speck
0.2	18/05/2018	Peer review	Grigorios Tzortzis
1.0	23/05/2018	Final version created	René Speck

Author List

Organization	Name	Contact Information
InfAI	Michael Röder	roeder@informatik.uni-leipzig.de
InfAI	René Speck	speck@informatik.uni-leipzig.de
InfAI	Felix Conrads	conrads@informatik.uni-leipzig.de
InfAI	Axel-Cyrille Ngonga Ngomo	ngonga@infai.org
InfAI	Ricardo Usbeck	usbeck@informatik.uni-leipzig.de
InfAI	Diego Moussallem	moussallem@informatik.uni-leipzig.de

Executive Summary

This deliverable presents the second version of the benchmark generator and benchmarking types for knowledge extraction frameworks. One of the key solutions presented herein is the Bengal generator, which allows creating large gold standard datasets for the evaluation of the runtime performance of named entity recognition, named entity linking, relation extraction and knowledge extraction solutions. In addition, we present variations on benchmarking (quality-focused and performance-focused) implemented by our suite.

The deliverable begins by giving a brief introduction to the goals and targets of the task. It then describes briefly the extensions of the Bengal generator. The different approaches that can be used for benchmarking named entity recognition, entity linking solutions, relation extraction and knowledge extraction are finally presented. We then give insights in the OKE challenge and how we applied the Hobbit platform within the challenge as well as the performance of the participated systems. An overview of baseline implementations and their performance within the Hobbit platform is finally given.

Contents

1 Introduction

This deliverable presents the second version of our suite of benchmarks for unstructured data. The suite was developed based on three considerations: *reuse*, *scalability* and *choke-point-based design*. The three considerations were introduced in detail in the first version of the suite.[1]

The result of this work package includes (1) a benchmark generator dubbed BENGAL, (2) a suite of 16 knowledge extraction benchmarks, 8 of these benchmarks were manually created and focus on the quality of the annotations that a benchmarked system can achieve, the other 8 benchmarks test the system behaviour when confronted with a high load and were generated using BENGAL, (3) the integration of 16 systems that are able to be benchmarked by our suit and (4) a baseline system for benchmarking dubbed Fox. Note that we can generate any number of benchmarks by configuring and running BENGAL on a knowledge base of choice.

The result of the second version of this work package includes (1) an extension of the benchmark generator BENGAL for creating new benchmarks described in this deliverable as well as for multilingual benchmark generation, (2) an extension of Fox with Ocelot to support the new benchmarks with a baseline system, (3) new benchmarks and (4) the integration of additional knowledge extraction systems to our suit.

This document is structured in the following way. In section 2, we present the BENGAL generator. Section 3 takes a holistic view of the work package and describes the general structure of the benchmarks, the different experiment types and the two different sets of benchmarks. In section 4 we give insights in the OKE challenge, the task descriptions and the results of the finished challenge. In section 5, the baseline implementations for single benchmarks are explained. We conclude the deliverable in section 6.

2 BENGAL

We described the BENGAL approach in detail in the first version of the deliverable for Named Entity Recognition (NER) and Named Entity Linking (NEL). In the second version we extended the approach to automatically generate Relation Extraction and Knowledge Extraction (cf. section 3.1.2 and section 3.1.3) benchmarks. Thus, with BENGAL we are able to generate benchmarks at scale for four types: Named Entity Recognition, Named Entity Disambiguation and Linking, Relation Extraction as well as Knowledge Extraction.

Additionally, BENGAL is prepared for multilingual benchmark generation. At the moment, we integrated Brazilian Portuguese language support and plan to integrate more in the future. We implemented BENGAL for Brazilian Portuguese relying on a Portuguese RDF verbalizer [10] and ran four multilingual NER and NEL frameworks thereon (MAG [11], DBpedia Spotlight [8], Babelfy [9], and PBOH [3]). In addition, we evaluated the performance of these annotators on subsets of HAREM[2] which is a manually created dataset[3]. While the extension of BENGAL to Portuguese is an important result in itself, our results also provide new insights in the NER and NEL performance of existing solutions. Amongst other, our results suggest that existing solutions are mostly biased towards a high precision but often achieve a lower recall on this language. For example, Spotlight's and Babelfy's recall remain below 0.6 in most cases while their precision goes up to 0.9. This clearly results from

[1] https://project-hobbit.eu/wp-content/uploads/2017/06/D3.2.1_First_Version_of_the_Data_Extraction_Benchmark_for_unstructured_data.pdf

[2] http://www.linguateca.pt/HAREM

[3] all Portuguese results can be found at http://faturl.com/bengalpt

the lack of training data for these resource-poor languages. In future work, we intend to quantify this phenomenon across other resource-poor languages and create datasets to push the development of tools to process these languages.

3 Benchmarking

In this section we describe the implemented and integrated benchmarks as well as the experiment types.

The benchmark suite implemented within HOBBIT reuses some of the concepts developed within the open-source project GERBIL. These concepts were migrated and adapted to the HOBBIT architecture. The implementation covers Named Entity Recognition (NER), Named Entity Disambiguation and Linking (NEL), Relation Extraction (RE) as well as Knowledge Extraction (KE). Correspondingly, the different components of our knowledge extraction benchmark abide by the specification described in D2.1.[4] A piece of the graphical user interface is depicted in fig. 1 that shows the benchmarks integrated into the platform.

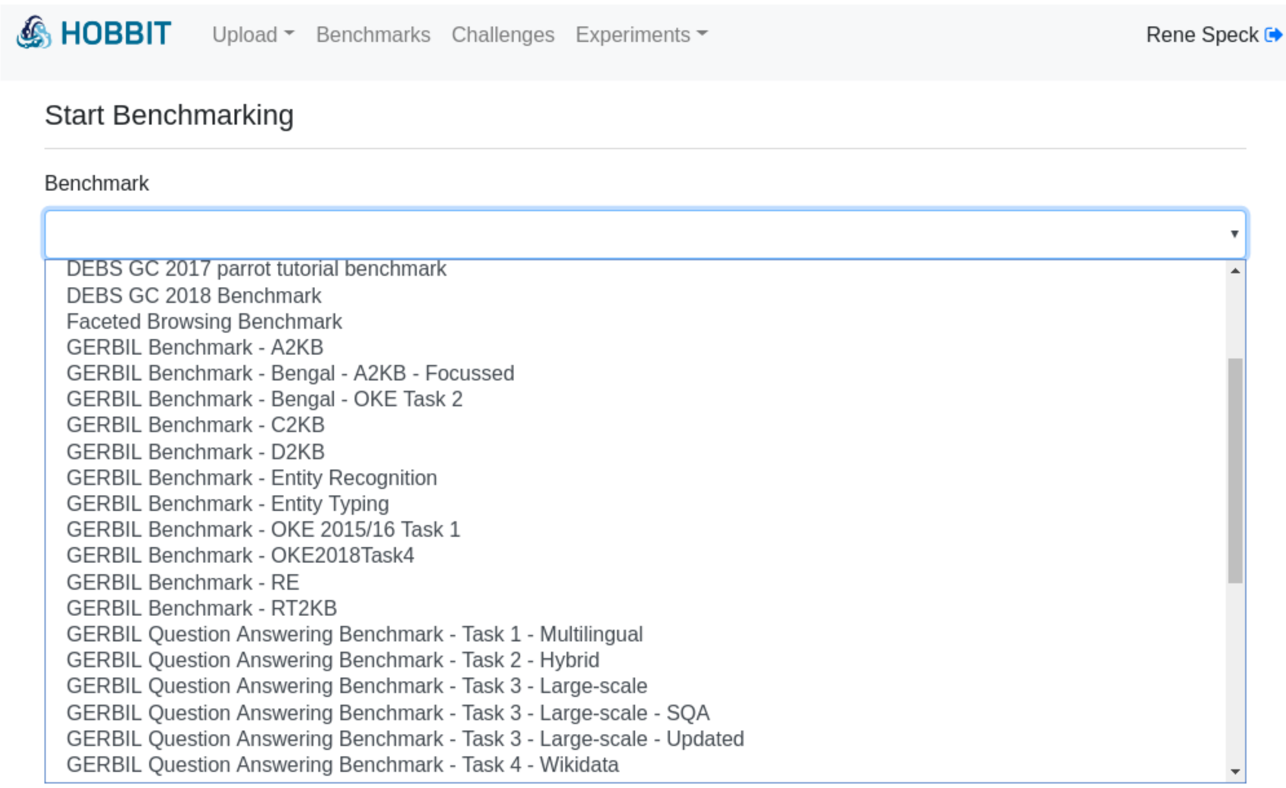


Figure 1: Part of the graphical user interface of the platform.

We provide two different implementations for the data and the task generators that are described in section 3.2 and section 3.3 in more detail. In general, the data generators send fully annotated documents to the task generators. The task generator forwards the fully annotated document to the evaluation storage but only the text of the document and the necessary meta data to the benchmarked

[4] This deliverable can be found at https://project-hobbit.eu/wp-content/uploads/2017/06/D3.2.1_First_Version_of_the_Data_Extraction_Benchmark_for_unstructured_data.pdf.

system. The benchmarked system performs the extraction task and sends its result to the evaluation storage. The evaluation module compares the annotations generated by the benchmarked system with the annotations that have been expected and calculates micro as well as macro values of precision, recall and F1-measure.

3.1 Experiment types

Table 1 briefly summarizes the extraction experiment types that are offered by the HOBBIT platform[5]. The first eight experiments are from the first version of the platform and the last two experiment types are new in the second version which we describe in detail.

Name	Input	Output
A2KB	plain text	the positions of entities inside the text and their URIs in a KB
C2KB	plain text	the URIs of entities that are important for that text
D2KB	plain text with already marked entities	the positions of entities inside the text and their URIs in a KB
Entity Typing	plain text with already marked entities	the URIs of the types of the entities
Entity Recognition	plain text	the positions of entities inside the text
OKETask1	plain text	the positions of entities inside the text, their URIs in a KB and the URIs of their types
OKETask2	plain text with a single marked entity and a description of its type	the positions and the URI of the type(s) of the given entity
RT2KB	plain text	the positions of entities and the URIs of their types
RE	plain text with marked entities, the URIs of the entities and the URIs of the types	Statements with URIs of the entities and the URI of the predicate
OKE2018Task4	plain text	the positions of entities inside the text, the URIs of the types of the entities, statements with URIs of the entities and the URI of the predicate.

Table 1: Overview of GERBIL experiment types migrated to the HOBBIT platform.

[5]The types are known from GERBIL. The interested reader is referenced to [15] for further details.

3.1.1 Preliminaries and Notation

This subsection defines terminologies and notations used in this deliverable to describe the experiments.

Knowledge Base

A knowledge base K consists of a set of entities E_K, an entity type hierarchy T_K with a function that maps each entity to its types $\psi_K : E_K \rightarrow T_K \times T_K \times \ldots$, a relation type hierarchy R_K with a function that maps each relation to its domain and range entity types $\phi_K : R_K \rightarrow T_K \times T_K$ and relation instances: $F_K = \{r(e_1, e_2)\} \subset R_K \times E_K \times E_K$.

Named Entity Identification

The identification of named entities in a given text D aims to find named entity mentions $M = \{m_i\}_{i=1,2,\ldots}$ that express named entities. A named entity mention m is a sequence of token in D identified by its start and end index $I_M = \{(a,b)_i\}_{i=1}^{|M|}$ where $a, b \in \mathbb{N}$ and $a < b$.

Named Entity Disambiguation and Linking

The aim of named entity disambiguation and linking to a knowledge base K is to assign each named entity mention $m \in M$ in D to an entity in K if possible, otherwise to generate a new resource for such an emerging entity, i.e., $\varphi : M \rightarrow E_K \cup E_{\bar{K}}$ is a function that maps an entity mention to an entity in E_K or, in case of an unlinkable mention, to a newly generated entity in $E_{\bar{K}}$ that not exists in K.

Closed Binary Relation Extraction

The aim of closed binary relation extraction is to find relations $r(e_j, e_k)$ expressed in a given text D with $r \in R_K$ and $e_j, e_k \in E_K \cup E_{\bar{K}}$. Often, closed binary relation extraction is limited to a subset of relations in K: $R \subset R_K$.

RDF/Turtle Prefixes

Listing 1 depicts the RDF/Turtle prefixes we use in all experiment example in- and outputs in this deliverable.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix aksw: <http://aksw.org/notInWiki/> .
@prefix oa: <http://www.w3.org/ns/oa#> .
```

Listing 1: Prefixes for the examples.

3.1.2 Benchmark — Relation Extraction (RE)

In the RE experiment, a benchmarking system gets documents, each consists of text together with annotations about the named entity mentions in the text and named entity linkings to the DBpedia knowledge base. A benchmarking system is expected to respond to such annotated documents with statements about relations that hold between the annotated named entities in the text.

Given documents with sentences D, the DBpedia knowledge base K, a target entity type hierarchy T with $T \subset T_K$ and a target relation type hierarchy R with $R \subset R_K$. Furthermore, annotations of the documents are given, i.e., entity mentions M with the positions I_M, the disambiguation function $\varphi : M \to E_K \cup E_{\bar{K}}$ and the types of entities $\psi_K : E_K \cup E_{\bar{K}} \to T$. The domain and range entity types of the relations in this task are given by a function of K, $\phi_K : R \to T \times T$.

The aim of this task is to find binary relations $r(e_j, e_k)$ with $r \in R$ where $e_j, e_k \in E_K \cup E_{\bar{K}}$.

Examples

Listing 2 shows an input example document of this experiment type and listing 3 shows the statements that have to be added to the input to provide the expected response document.

```
<http://www.ontologydesignpatterns.org/data/oke-challenge-2018/task-3/sentence-1#char=0,78>
  a nif:RFC5147String , nif:String , nif:Context ;
  nif:beginIndex "0"^^xsd:nonNegativeInteger ;
  nif:endIndex "78"^^xsd:nonNegativeInteger ;
  nif:isString "Conor McGregor's longtime trainer, John Kavanagh, is ready to shock the world."^^xsd:string .

<http://www.ontologydesignpatterns.org/data/oke-challenge-2018/task-3/sentence-1#char=0,22>
  a nif:RFC5147String , nif:String , nif:Phrase ;
  nif:anchorOf "Conor McGregor's"^^xsd:string ;
  nif:beginIndex "0"^^xsd:nonNegativeInteger ;
  nif:endIndex "22"^^xsd:nonNegativeInteger ;
  nif:referenceContext <http://www.ontologydesignpatterns.org/data/oke-challenge-2018/task-3/sentence-1#char=0,78>
      ;
  its:taClassRef dbo:Person ;
  itsrdf:taIdentRef dbr:Conor_McGregor .

<http://www.ontologydesignpatterns.org/data/oke-challenge-2018/task-3/sentence-1#char=35,48>
  a nif:RFC5147String , nif:String , nif:Phrase ;
  nif:anchorOf "John Kavanagh"^^xsd:string ;
  nif:beginIndex "35"^^xsd:nonNegativeInteger ;
  nif:endIndex "48"^^xsd:nonNegativeInteger ;
  nif:referenceContext <http://www.ontologydesignpatterns.org/data/oke-challenge-2018/task-3/sentence-1#char=0,78>
      ;
  its:taClassRef dbo:Person ;
  itsrdf:taIdentRef aksw:John_Kavanagh .
```

Listing 2: Example request document.

```
[]
  a rdf:Statement , oa:Annotation ;
  rdf:object dbr:Conor_McGregor ;
  rdf:predicate dbo:trainer ;
  rdf:subject aksw:John_Kavanagh ;
  oa:hasTarget [
    a oa:SpecificResource;
    oa:hasSource <http://www.ontologydesignpatterns.org/data/oke-challenge-2018/task-3/sentence-1#char=0,78> ] .
```

Listing 3: Example response.

3.1.3 Benchmark — Knowledge Extraction (OKE2018Task4)

In the OKE2018Task4 for knowledge extraction, a benchmarking system gets documents with plain text only. A benchmarking system is expected to respond with annotations about the named entity mentions, entity links to DBpedia and with statements about relations that hold between the annotated named entities in the text. Basically, this experiment is a combination of the experiments: Entity Recognition, D2KB and RE.

Example

Listing 4 shows an input example document of this experiment type and listing 5 shows the named entity mentions, positions, linkings and statements that have to be added to the input to provide the expected response document.

```
<http://www.ontologydesignpatterns.org/data/oke-challenge-2018/task-4/sentence-1#char=0,78>
  a nif:RFC5147String , nif:String , nif:Context ;
  nif:beginIndex "0"^^xsd:nonNegativeInteger ;
  nif:endIndex "78"^^xsd:nonNegativeInteger ;
  nif:isString "Conor McGregor's longtime trainer, John Kavanagh, is ready to shock the world."^^xsd:string .
```

Listing 4: Example request document.

```
<http://www.ontologydesignpatterns.org/data/oke-challenge-2018/task-4/sentence-1#char=0,22>
  a nif:RFC5147String , nif:String , nif:Phrase ;
  nif:anchorOf "Conor McGregor's"^^xsd:string ;
  nif:beginIndex "0"^^xsd:nonNegativeInteger ;
  nif:endIndex "22"^^xsd:nonNegativeInteger ;
  nif:referenceContext <http://www.ontologydesignpatterns.org/data/oke-challenge-2018/task-4/sentence-1#char=0,78>
      ;
  itsrdf:taIdentRef dbr:Conor_McGregor .

<http://www.ontologydesignpatterns.org/data/oke-challenge-2018/task-4/sentence-1#char=35,48>
  a nif:RFC5147String , nif:String , nif:Phrase ;
  nif:anchorOf "John Kavanagh"^^xsd:string ;
  nif:beginIndex "35"^^xsd:nonNegativeInteger ;
  nif:endIndex "48"^^xsd:nonNegativeInteger ;
  nif:referenceContext <http://www.ontologydesignpatterns.org/data/oke-challenge-2018/task-4/sentence-1#char=0,78>
      ;
  itsrdf:taIdentRef aksw:John_Kavanagh .

[]
  a rdf:Statement , oa:Annotation ;
  rdf:object dbr:Conor_McGregor ;
  rdf:predicate dbo:trainer ;
  rdf:subject aksw:John_Kavanagh ;
  oa:hasTarget [
    a oa:SpecificResource;
    oa:hasSource <http://www.ontologydesignpatterns.org/data/oke-challenge-2018/task-4/sentence-1#char=0,78> ] .
```

Listing 5: Example response document.

3.2 Quality-focused benchmarking (Scenario A)

The first type of benchmarking provided by our suite focuses on the measurement of the quality a system achieves on a given set of documents. We assume that each benchmark consists of a set of documents. First, the platform orders documents from the benchmark. The documents are sent to the benchmarked system one at a time. The benchmarked system generates a response and sends it to the evaluation storage before receiving the next document. That means that the benchmarked system can be configured to concentrate all its resources on a single request and does not need to scale to a large number of requests.

In this approach to benchmarking, we mainly rely on manually created gold standards, however we could use Bengal datasets as well. Table 2 lists the corpora that are currently available for benchmarking within Hobbit. Note that we could not migrate all datasets available in GERBIL to the Hobbit platform due to licensing reasons.[6] To compensate, we created 7 new datasets, see table 3.

This benchmarking approach evaluates the quality in terms of precision, recall and F-measure.

[6] A list of datasets available in GERBIL together with their licenses can be found at https://github.com/AKSW/gerbil/wiki/Licences-for-datasets.

	A2KB	C2KB	D2KB	ER	ET	OKE18 Task4	OKE15/16 Task1	RE	RT2KB
ACE2004	✓	✓	✓	✓					
DBpediaSpotlight	✓	✓	✓	✓					
Derczynski	✓	✓	✓	✓					
ERD2014	✓	✓	✓	✓					
GERDAQ-Dev	✓	✓	✓	✓					
GERDAQ-Test	✓	✓	✓	✓					
GERDAQ-TrainingA	✓	✓	✓	✓					
GERDAQ-TrainingB	✓	✓	✓	✓					
IITB	✓	✓	✓	✓					
Kore50	✓	✓	✓	✓					
MSNBC	✓	✓	✓	✓					
N3-Reuters-128	✓	✓	✓	✓					
N3-RSS-500	✓	✓	✓	✓					
OKE15 Task1 evaluation dataset	✓	✓	✓	✓	✓		✓		✓
OKE15 Task1 example set	✓	✓	✓	✓	✓		✓		✓
OKE15 Task1 gold standard sample	✓	✓	✓	✓	✓		✓		✓
OKE16 Task1 evaluation dataset	✓	✓	✓	✓	✓		✓		✓
OKE16 Task1 example set	✓	✓	✓	✓	✓		✓		✓
OKE16 Task1 gold standard sample	✓	✓	✓	✓	✓		✓		✓
OKE17 Task1 evaluation	✓		✓				✓		✓
OKE17 Task1 training	✓		✓				✓		✓
OKE17 Task2 evaluation	✓		✓				✓		✓
OKE17 Task2 training	✓		✓				✓		✓
OKE17 Task3 evaluation	✓		✓				✓		✓
OKE17 Task3 training	✓		✓				✓		✓
OKE18 Task1 dataset	✓								
OKE18 Task2 dataset	✓								
OKE18 Task3 dataset								✓	
OKE18 Task4 dataset						✓			
Ritter				✓	✓				✓
Senseval 2				✓					
Senseval 3				✓					
UMBC-Test				✓	✓				✓
UMBC-Train				✓	✓				✓
WSDM 2012		✓							

Table 2: Overview over datasets and their experiment types available for benchmarking in the HOBBIT platform.

3.3 Performance-focused benchmarking (Scenario B)

The second approach to benchmarking implemented by our approach aims to stress test the benchmarked system and to evaluate its runtime and quality in terms of precision, recall and F-measure. This approach hence focuses on the ability of a system to annotate documents in parallel with an

Name	Task	Type	License	Language
OKE17 Task1	OKE2015 Task 1	news	-	en
OKE17 Task2	OKE2015 Task 1	news	-	en
OKE17 Task3	OKE2015 Task 1	music	-	en
OKE18 Task1	OKE2015 Task 1	news	-	en
OKE18 Task2	OKE2015 Task 1	news	-	en
OKE18 Task3	RE	news	-	en
OKE18 Task4	OKE2018 Task 4	news	-	en

Table 3: Overview of new GERBIL gold corpora that are available for benchmarking in the HOBBIT platform for a specific task. The content type it given in the table as well as the license and the language of the corpora.

increasing amount of load. The data generators create a large amount of synthetic documents from a given KB using BENGAL[7]. These documents are sent to the system in parallel without waiting for responses for previous requests.[8] The data generators go through a number of **phases**. In each phase, a predefined delay between the sending of two documents is set and used. After a phase has been completed, i.e., after a predefined number of documents per phase have been sent by the data generator, the next phase starts with half of the waiting time of the previous phase. This simulates a workload that becomes higher from phase to phase until no time is left between sending two documents.

Let $p \geq 2$ be the number of phases and $t_1, ..., t_p$ the delays between sending two documents. These delays are calculated using the following equations.

$$t_p = 0ms \tag{1}$$
$$t_{p-1} = 62.5ms \tag{2}$$
$$t_i = 2 \times t_{i+1} \tag{3}$$
$$\tag{4}$$

where $0 < i < (p-1)$. The parameters for these benchmarks are

- The KB used for generating the documents
- The set of entity types used for generation
- The number of data generators
- The number of phases

[7] https://github.com/aksw/bengal

[8] To send messages in parallel, the number of data generators needs to be > 1.

- The number of documents per phase per data generator

For evaluating the performance of a system, we introduced the following β measure

$$\beta = \frac{f}{r} \tag{5}$$

where f is the F1-score that has been achieved on the given document and r is the runtime in seconds. For every phase as well as for all phases an average β can be used

$$\bar{\beta} = \frac{1}{|D|} \sum_{d \in D} \frac{f_d}{r_d} \tag{6}$$

where D is the set of documents used during a phase or during the whole benchmark.

4 Open Knowledge Extraction (OKE) Challenge 2017/2018

4.1 Introduction

The Open Knowledge Extraction Challenge invites researchers and practitioners from academia as well as industry to compete to the aim of pushing further the state of the art in knowledge extraction from text for the Semantic Web. The challenge has the ambition to provide a reference framework for research in this field by redefining a number of tasks typically from information and knowledge extraction by taking into account Semantic Web requirements and has the goal to test the performance of knowledge extraction systems. This year, the challenge goes in the fourth round and consists of four tasks which include named entity identification, disambiguation by linking to a knowledge base as well as relation and knowledge extraction. The challenge makes use of small gold standard datasets that consist of manually curated documents and large silver standard datasets that consist of automatically generated synthetic documents. The performance measure of a participating system is twofold base on (1) Precision, Recall, F1-measure for quality-focused benchmarking and on (2) Precision, Recall, F1-measure with respect to the runtime of the system for performance-focused benchmarking.

4.2 OKE Challenge 2017/2018

For both challenges we use the HOBBIT platform and the integrated benchmarks to benchmark the systems of the participants.

The OKE Challenge 2017 took place at the ESWC 2017 from May 28th, 2017 to June 1st, 2017 in Portoroz, Slovenia.[9] Thus, the challenge finished and the results are available.

The OKE Challenge 2018 takes place at the ESWC 2018 from June 3rd, 2018 to June 7th, 2018 in Crete, Greece.[10] Thus the challenge is currently running.

In the next sections we describe the OKE Challenge 2017 tasks of as well as briefly the results of the finished challenge. The detailed description and results are available in [18]. In this challenge two systems participated, ADEL [14] and the baseline system Fox [17].

[9] https://2017.eswc-conferences.org/call-challenges
[10] https://2018.eswc-conferences.org/call-for-challenges

4.3 OKE Challenge 2017 Task 1: Focused Named Entity Identification and Linking

The first task comprises a two-step process with the identification of named entities in sentences and the disambiguation of the identified entities by linking these entities to resources in the given knowledge base. A competing system is expected to identify named entity mentions in a given document as well as the start and end indices of the entity mentions, further to generate a URI to link each mention if possible, otherwise to generate a URI for an emerging entity. This task is limited to a subset of entity types and their associated subtypes[11], i.e., $T := \{\text{dbo:Person}, \text{dbo:Place}, \text{dbo:Organisation}\}$.

4.3.1 Results

The results of the OKE Challenge 2017 challenge task 1 are provided in table 4 for the quality-focused benchmarking and for the performance-focused benchmarking in fig. 2a.

Table 4: Results on task 1 of OKE Challenge 2017.

Experiment Type	Micro measures	Scenario A		Scenario B	
		Adel	Fox	Adel	Fox
A2KB	Precision	33.24	53.61	18.28	59.12
	Recall	30.18	46.72	22.36	72.51
	F1-measure	31.64	**49.93**	20.12	**65.15**
Recognition	Precision	91.62	92.47	74.39	73.27
	Recall	83.20	80.58	90.98	89.85
	F1-measure	**87.21**	86.12	**81.85**	80.72
D2KB	Precision	40.15	61.96	28.03	93.87
	Recall	27.82	41.47	19.26	66.99
	F1-measure	32.87	**49.69**	22.83	**78.19**
	Time	7.98	6.98	231.31	179.29
	Errors	0	0	6	1

4.4 OKE Challenge 2017 Task 2: Broader Named Entity Identification and Linking

This task extends the former task towards the entity types. Beside the three types of the first task, a competing system might have to identify other types of entities and to link these entities as well. In the first column in table 5, a complete list of types that are considered in this task is provided. The

[11] The complete type hierarchy: http://mappings.dbpedia.org/server/ontology/classes/.

middle column contains example subtypes of the corresponding type if any such type is available and the last column contains example instances in the knowledge base for the related types respectively subtypes.

Table 5: Types, subtype examples and instance examples.

Type	Subtypes	Instances
`Activity`	`Game`, `Sport`	`Baseball,Chess`
`Agent`	`Organisation`, `Person`	`Leipzig_University`
`Award`	`Decoration`, `NobelPrize`	`Humanitas_Prize`
`Disease`		`Diabetes_mellitus`
`EthnicGroup`		`Javanese_people`
`Event`	`Competition`, `PersonalEvent`	`Battle_of_Leipzig`
`Language`	`ProgrammingLanguage`	`English_language`
`MeanOfTransportation`	`Aircraft`, `Train`	`Airbus_A300`
`PersonFunction`	`PoliticalFunction`	`PoliticalFunction`
`Place`	`Monument`, `WineRegion`	`Beaujolais`, `Leipzig`
`Species`	`Animal`, `Bacteria`	`Cat`, `Cucumibacter`
`Work`	`Artwork`, `Film`	`Actrius`, `Debian`

4.4.1 Results

The results of the OKE Challenge 2017 challenge task 2 are provided in table 6 for the quality-focused benchmarking (scenario A) and for the performance-focused benchmarking (scenario B). In fig. 2b, the measured beta for the performance-focused benchmarking is depicted.

4.5 OKE Challenge 2017 Task 3: Focused Musical Named Entity Recognition and Linking

Task 3 is composed of two subtask, 3A and 3B. Task 3A focuses on Musical Named Entity Recognition of named entities related to three different types and task 3B is focuses on linking those named entities to the knowledge base.

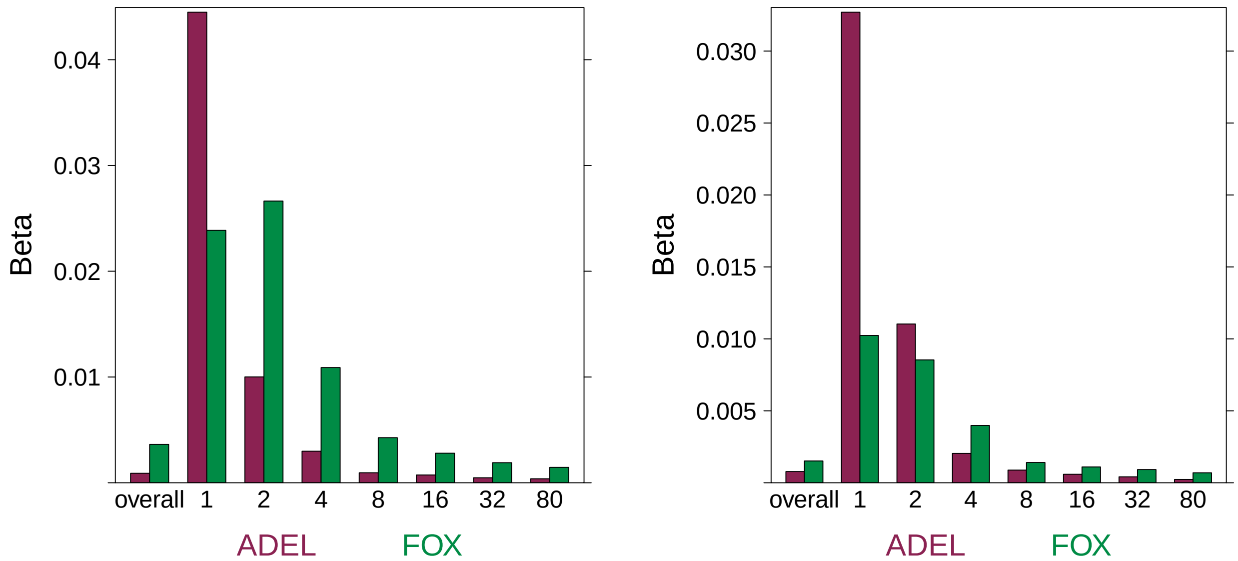
In the following, we first summarize the results on the first subtask and then on the second.

4.5.1 Task 3A

The measured values for task 3A are depicted in table 7.

Table 6: Results on task 2.

Experiment Type	Micro measures	Scenario A		Scenario B	
		ADEL	FOX	ADEL	FOX
A2KB	Precision	31.40	56.15	17.44	44.90
	Recall	28.14	38.53	18.93	39.83
	F1-measure	29.68	**45.70**	18.15	**42.22**
Recognition	Precision	87.68	95.90	72.31	74.64
	Recall	78.57	65.80	78.50	66.21
	F1-measure	**82.88**	78.05	**75.27**	70.17
D2KB	Precision	39.93	63.42	28.57	82.38
	Recall	25.76	35.28	17.47	36.92
	F1-measure	31.32	**45.34**	21.68	**51.00**
	Time	4.60	7.66	261.48	245.99
	Errors	0	1	57	0



(a) β on Task 1.

(b) β on Task 2.

Figure 2: β values on several phases and overall.

4.5.2 Task 3B

The measured values for task 3.2 are depicted in table 8.

Table 7: Results on Task 3A.

Experiment Type	Micro measures	ADEL	FOX
RT2KB	Precision	26.99	0
	Recall	27.24	0
	F1-measure	**27.12**	0
Recognition	Precision	35.03	63.02
	Recall	74.57	49.21
	F1-measure	47.66	**55.27**
Typing	Precision	64.33	0
	Recall	64.91	0
	F1-measure	**64.62**	0
	Time	37.19	7.82
	Errors	16	0

Table 8: Results on Task 3B

Experiment Type	Micro measures	ADEL	FOX
D2KB	Precision	6.82	10.10
	Recall	5.10	4.97
	F1-measure	5.83	**6.66**
	Time	36.96	9.15
	Errors	16	0

5 Benchmarking Baseline

The baseline system implementations can be separated into two parts. First, we migrated the web service adapters implemented in the GERBIL project to the HOBBIT architecture for those web services that are free to use. However, it needs to be pointed out that these web services are not hosted in the HOBBIT platform but on a different server. Thus, they can only be used for a quality focused benchmark as they are executed by the GERBIL benchmarking platform. Their usage in a performance focused benchmark would create a huge workload for this publicly available web service and might be even seen as a denial of service attack. Table 9 lists the available web services that can be used as baseline systems.

For the performance focussed benchmarks, we implemented a system adapter for our open source framework[12] Fox [16].

In the following subsections we describe our intern benchmarking baseline approach. It consists of Fox, for named entity recognition, of Agdistis [19], for named entity disambiguation and linking, as well as of Ocelot, for relation extraction. Ocelot is a new system for relation extraction and currently under review. The three tools are integrated in one framework, Fox, that serves as benchmarking baseline system. Fox without Ocelot was described in the first version of the deliverable, thus in this version we describe Ocelot for relation extraction combined with Fox for knowledge extraction in the following subsections.

	A2KB	C2KB	D2KB	ER	ET	OKE18 Task4	OKE15/16 Task1	RE	RT2KB
[19] AGDISTIS			✓						
[7] AIDA	✓	✓	✓	✓					
[9] Babelfy	✓	✓	✓	✓					
[8] Spotlight for A2KB	✓								
[8] Spotlight	✓	✓	✓	✓	✓		✓		✓
[1] Dexter	✓	✓	✓	✓					
[21] DoSeR			✓						
[16] FOX	✓	✓	✓	✓	✓	✓	✓	✓	✓
[16] FOX Balie	✓	✓	✓	✓	✓		✓		✓
[16] FOX Illinois	✓	✓	✓	✓	✓		✓		✓
[16] FOX OpenNLP	✓	✓	✓	✓	✓		✓		✓
[16] FOX Stanford	✓	✓	✓	✓	✓		✓		✓
[2] FRED	✓	✓	✓	✓	✓		✓		✓
FREME NER	✓	✓	✓	✓	✓		✓		✓
[6] NERFGUN			✓						
[4] PBOH			✓						
[13] WAT	✓	✓	✓	✓					
[20] xLisa using NER	✓	✓	✓	✓					
[20] xLisa using NGRAM	✓	✓	✓	✓					

Table 9: Overview of implemented adapters of annotation web services.

5.1 Ocelot

Ocelot is a closed relation extraction approach based on distance supervision by using distributed semantics and a tree generalization process. It extracts tree sets from a corpus where each set consists of trees which express a specific relation utilized from a knowledge base. The generalized trees are applicable in a plethora of applications to mine relationships in a precise way, particularly by its non overlapping characteristic to multiple relations as it is the case in many common state-of-the-art systems like Boa [5] and Patty [12]. Figure 3 depicts the general overview of Ocelot's data flow.

Ocelot starts by preprocessing the corpus with natural language processing tools to gain linguistic annotations and to store those annotations in the Solr index. Then, Ocelot queries the knowledge

[12] http://aksw.org/Projects/FOX.html

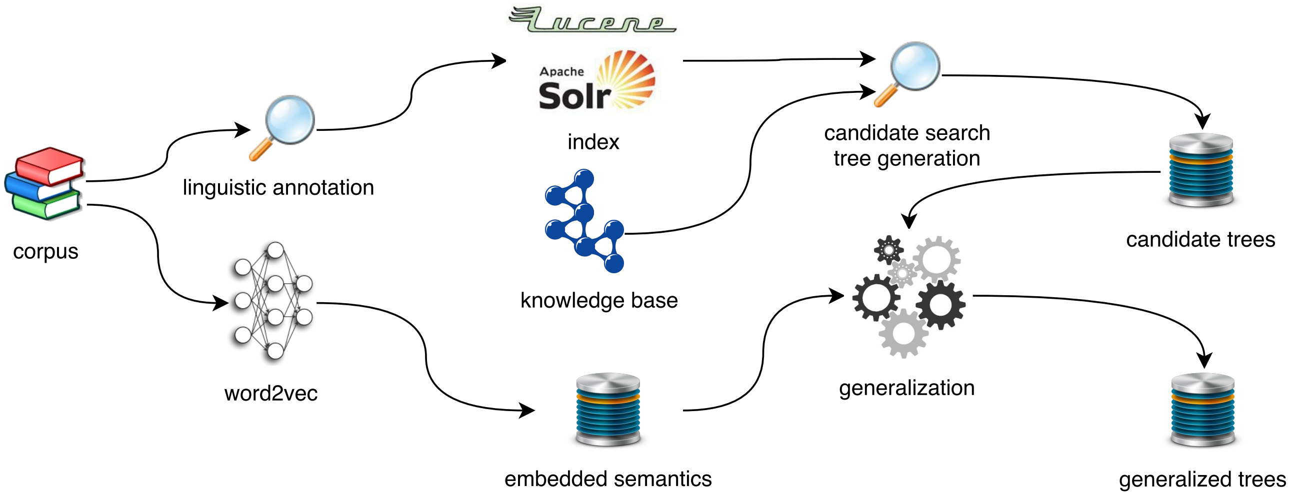


Figure 3: Overview of Ocelot's data flow.

base for predicates and instances that serve as target relations as well as relation instances in our approach. After that, Ocelot retrieves labels of the instances that are used to search in the index to find candidate sentences. Ocelot creates dependency parse trees on the candidate sentences and stores them. In the generalization step, Ocelot starts with creating embedded semantics. The embedded semantics together with labels from the knowledge base for the predicates as well as from other sources like Wordnet are used in the generalization step.

Linguistic Annotation

This module removes markup language from the corpus, splits articles to sentences and filters sentences by length. Furthermore, it creates linguistic annotations for each sentence and stores the annotations in the index for fast search.

Candidate Selection

The candidate selection module's main function is to select candidate sentences from the index that at least contain two named entities. With this condition, we expect a relation between the named entities. The module uses background knowledge created from the knowledge base to search for the candidate sentences. For each selected candidate sentence it generates its dependency parse tree.

Embedded Semantics

This module is a preprocessing step for the generalization. We create word embeddings on the corpus and use predicate labels from the background knowledge to find semantically similar words in several sources.

Generalization

The generalization module takes as input the candidate trees as well as the results of the embedded semantics module to generalize, score and rank the candidate trees.

5.2 Evaluation

We evaluated Ocelot two-fold, qualitatively and quantitatively, on the English Wikipedia and utilized the knowledge base DBpedia. We first evaluated our approach quantitatively with the measure of statistical variability, precision, with and without filtering trees by the Embedded Semantics module. Then, we compared the results of our approach with the results of two common state-of-the-art systems, Patty and Boa. Therefore, we compared precision, recall and F1-Score of the three systems. In the qualitative evaluation, we compared the quality of the extraction of relations from natural language text of our system with the extraction results of Boa.

5.2.1 Quantitative Evaluation

We first compared the results of our approach with (F) and without filtered (NF) trees by the Embedded Semantics module. The results are given in table 10. For the top ranked trees, k=1, we observed a precision of 58.17% without filtering and 94.74% with filtering trees. We compared the top 1–5 ranked trees and observed on each a significant increase of precision. We also observed a decrease of the number of trees by filtering from 55 trees to 19 trees. Because of the significant increase of the precision, we decided to filter the trees by the module.

Table 10: Precision and the number of trees without filter (NF) and with filter (F) for the top k ranked trees.

top k	NF		F	
	# trees	P	# trees	P
1	55	58.18	19	94.74
2	102	57.84	30	93.33
3	143	57.34	40	95.00
4	182	54.95	47	93.62
5	219	55.25	54	94.44

We manually compared the pattern[13] of Boa and Patty with the trees of our approach. To compare trees with pattern we created pattern representations out of our trees. We were not able to evaluate the results with those from Patty in an automated way, because it is not supported.[14] Table 11 depicts the results of our evaluation. We manually assessed the pattern for each tool with Precision, Recall and F-Score. We compared the top k extraction results for four relations supported by all three systems, `dbo:spouse`, `dbo:birthPlace`, `dbo:deathPlace` and `dbo:subsidiary`. For k we chose 1–5 and it turned out that our approach reached higher values on all five k.

[13] A pattern in Boa and Patty expresses a natural language representation of a relation and can consist of POS tags, lemmas and other generalized parts.

[14] We asked the authors for an API or the database with pattern.

Table 11: Precision, Recall and F-Score averaged over `dbo:spouse`, `dbo:birthPlace`, `dbo:deathPlace` and `dbo:subsidiary` for the top k pattern.

top k	Boa P/R/F1	Patty P/R/F1	Ocelot P/R/F1
1	75.00/8.120/14.58	75.00/9.550/16.67	**100.0/13.12/22.92**
2	62.50/12.66/20.94	62.50/15.39/24.24	**87.50/21.23/33.64**
3	58.33/18.51/27.86	66.67/24.94 /35.36	**91.67/34.35/48.93**
4	56.25/23.05/32.42	62.50/29.48/38.99	**91.67/40.19/54.73**
5	60.00/32.60/41.46	60.00/34.03/42.29	**86.67/43.77/56.55**

5.2.2 Qualitative Evaluation

We compared the quality of the extraction results of our approach with the extraction results of the state-of-the-art tool Boa. For the relation extraction with Boa, we chose the top 10 pattern for each relation from the Boa index. We compared Boa and Ocelot on the first 100 sentences of the top 3 viewed articles about persons in Wikipedia. The results are given in table 12, we replaced named entity mentions with its types, e.g. `Person`, since this is the preprocessing step for both tools. The ✕ indicates that the system found no relation in the sentence. The bold marked relation indicates correct extractions.

Table 12: Example relation extraction with Boa and Ocelot.

Examples	Boa	Ocelot
(`Person`) and his wife (`Person`)	✕	**dbo:spouse**
(`Person`) and (`Person`) were married	✕	**dbo:spouse**
(`Person`) met (`Person`)	dbo:spouse	dbo:spouse
(`Person`) was born in (`Place`)	dbo:deathPlace **dbo:birthPlace**	**dbo:birthPlace**
(`Person`) was born in 1905 in (`Place`)	✕	**dbo:birthPlace**
(`Person`) returned to (`Place`)	dbo:deathPlace dbo:birthPlace	✕
(`Person`) moved to (`Place`)	dbo:deathPlace dbo:birthPlace	✕

With Boa, we were able to extract one correct relation in one sentence. The sentence contains the pattern "(`Person`) was born in (`Place`)" for the relation `dbo:birthPlace` and was found by Boa, but this system found also a incorrect relation on the same sentence, `dbo:deathPlace`. In total, Boa extracted one correct relation and six incorrect relations. In three cases Boa extracted multiple incorrect relations in one sentence. In total With Ocelot, we extracted four correct relations and just one incorrect relation.

With Boa and Patty, we could not extract relations in sentences with pattern that hold not only in between named entities, e.g. "(`Person`) married (`Person`)", but also surrounding the named entities, e.g. "(`Person`) and (`Person`) were married". With Ocelot we were able to extract relations in such sentences (cf. table 12).

6 Conclusion

In this report, we presented the benchmarks and benchmarking approaches developed to evaluated Named Entity Recognition, Named Entity Linking, Relation Extraction and Knowledge Extraction for Big Linked Data. In particular, we presented Bengal, a benchmark generator that can generate annotated large-scale natural language benchmarks. We showed that the benchmarks generate by Bengal are diverse in their structure and can thus emulate a number of manually created benchmarks. During the benchmark creation process, we noticed that the quality of the benchmarks created by Bengal is actually superior to that of manually created benchmarks, which tend to contain errors of different types. We have hence begun and will continue developing benchmark cleaning approaches to deliver high-quality benchmarks through Hobbit. We have also presented how our performance benchmarks are implemented within the Hobbit platform. The choke-point-driven design of the suite has already allowed us to detect potential weaknesses of existing solutions.

References

[1] Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. Dexter: an open source framework for entity linking. In *Proceedings of the sixth international workshop on Exploiting semantic annotations in information retrieval*, 2013.

[2] Sergio Consoli and DiegoReforgiato Recupero. Using FRED for Named Entity Resolution, Linking and Typing for Knowledge Base Population. In *Semantic Web Evaluation Challenges*, volume 548 of *Communications in Computer and Information Science*, pages 40–50. Springer International Publishing, 2015.

[3] Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 927–938, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[4] Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 927–938, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[5] Daniel Gerber and Axel-Cyrille Ngonga Ngomo. Bootstrapping the linked data web. In *1st Workshop on Web Scale Knowledge Extraction @ ISWC 2011*, 2011.

[6] Sherzod Hakimov, Hendrik ter Horst, Soufian Jebbara, Matthias Hartung, and Philipp Cimiano. Combining textual and graph-based features for named entity disambiguation using undirected probabilistic graphical models. In *Proceedings of 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, 2016.

[7] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust Disambiguation of Named Entities in Text. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, Edinburgh, Scotland*, pages 782–792, 2011.

[8] Pablo N. Mendes, Max Jakob, Andres Garcia-Silva, and Christian Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *7th International Conference on Semantic Systems (I-Semantics)*, pages 1–8, 2011.

[9] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity Linking meets Word Sense Disambiguation: A Unified Approach. *TACL*, 2014.

[10] Diego Moussallem, Thiago Castro Ferreira, Marcos Zampieri, Maria Claudia Cavalcanti, Geraldo Xexeo, Mariana Neves, and Axel-Cyrille Ngonga Ngomo. RDF2PT: Generating Brazilian Portuguese Texts from RDF Data. In *LREC*, 2018.

[11] Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. MAG: A Multilingual, Knowledge-base Agnostic and Deterministic Entity Linking Approach. In *K-CAP 2017: Knowledge Capture Conference*, page 8. ACM, 2017.

[12] Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. Patty: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1135–1145, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[13] Francesco Piccinno and Paolo Ferragina. From TagME to WAT: a new entity annotator. In *Proceedings of the first international workshop on Entity recognition & disambiguation*, 2014.

[14] Julien Plu, Raphaël Troncy, and Giuseppe Rizzo. ADEL@OKE 2017: A generic method for indexing knowlege bases for entity linking. In *ESWC 2017, 14th European Semantic Web Conference, Open Extraction Challenge, 28th May-1st June 2017, Portoroz, Slovenia*, Portoroz, SLOVENIA, 05 2017.

[15] Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. GERBIL – Benchmarking Named Entity Recognition and Linking Consistently. *Under review for the Semantic Web Journal*, 2017.

[16] René Speck and Axel-Cyrille Ngonga Ngomo. Ensemble learning for named entity recognition. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig A. Knoblock, Denny Vrandecic, Paul T. Groth, Natasha F. Noy, Krzysztof Janowicz, and Carole A. Goble, editors, *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, volume 8796 of *Lecture Notes in Computer Science*, pages 519–534. Springer, 2014.

[17] René Speck and Axel-Cyrille Ngonga Ngomo. Ensemble learning for named entity recognition. In *The Semantic Web – ISWC 2014*, volume 8796 of *Lecture Notes in Computer Science*, pages 519–534. Springer International Publishing, 2014.

[18] René Speck, Michael Röder, Sergio Oramas, Luis Espinosa-Anke, and Axel-Cyrille Ngonga Ngomo. Open knowledge extraction challenge 2017. In *Semantic Web Evaluation Challenge*, pages 35–48. Springer International Publishing, 2017.

[19] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, SandroAthaide Coelho, Sören Auer, and Andreas Both. AGDISTIS - Graph-Based Disambiguation of Named Entities Using Linked Data. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble, editors, *The Semantic Web – ISWC 2014*, volume 8796 of *Lecture Notes in Computer Science*, pages 457–471. Springer International Publishing, 2014.

[20] Lei Zhang and Achim Rettinger. X-lisa: Cross-lingual semantic annotation. *PVLDB*, 7(13):1693–1696, 2014.

[21] Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. *DoSeR - A Knowledge-Base-Agnostic Framework for Entity Disambiguation Using Semantic Embeddings*, pages 182–198. Springer International Publishing, Cham, 2016.