Developing a Sustainable Platform for Entity Annotation Benchmarks

Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo

University of Leipzig, Germany
{roeder,usbeck,ngonga}@informatik.uni-leipzig.de

Abstract. The existing entity annotation systems that drive the extraction of RDF from unstructured data are hard to compare as their evaluation relies on different data sets and measures. We developed GERBIL, an evaluation framework for semantic entity annotation that provides developers, end users and researchers with easy-to-use interfaces for the agile, fine-grained and uniform evaluation of 9 annotation tools on 11 different data sets within 6 different experimental settings on 6 different measures. In this paper, we present the developed interfaces, data flows and data structures. Moreover, we show how GERBIL supports a better reproducibility and archiving of experimental results.

1 Introduction

The need for extracting structured data from text has led to the development of a large number of tools dedicated to the extraction of structured data from unstructured data (see [6] for an overview). While these tools do provide evaluation results, these results are rarely fully comparable as they commonly rely on different data sets or different measures. This is partly due to data preparation being a tedious problem in the annotation domain due to the different formats of the gold standards as well as the different data representations across reference data sets. Recently, benchmarking frameworks such as the BAT-framework [3] or NERD-ML [5] for entity annotation systems have began addressing the problem on reproducible experiments for entity annotation. With GERBIL[1] we aim to unify experiment setups, ease implementation and testing effort as well as contribute to an open, repeatable, publishable and archivable open science area to foster an active community of entity annotation tool developers.

GERBIL goes beyond the state of the art by extending the BAT-framework [3] as well as Nerd-ML [5] in several dimensions. In particular we provide fine-grained diagnostics for annotation tools, enhanced reproducibility through URIs for experiments, easily publishable results by providing results both as RDF (for machines) and tables (for humans). Overall, we provide the following features:

Feature 1: Extensible experiment types. An experiment type defines the way used to solve a certain problem when extracting information. GERBIL extends the six experiment types provided by the BAT framework [3] (including entity recognition and disambiguation) towards more general, URI based experiments. With this extension,

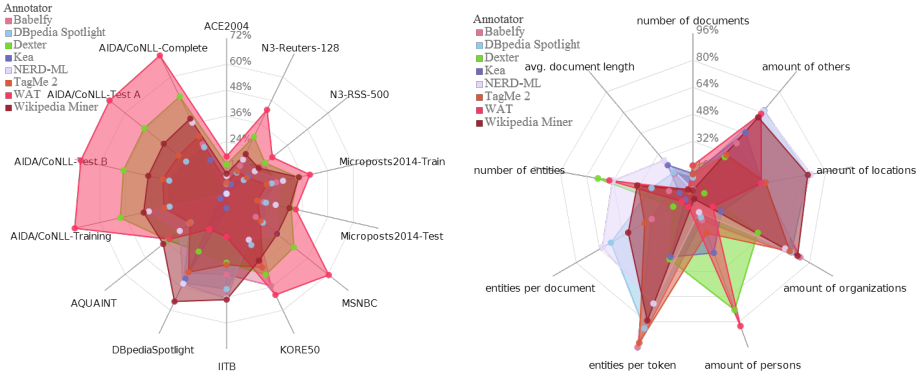
[1] More information and a demo can be found at http://gerbil.aksw.org

our framework can deal with gold standard data sets and annotators that link to any knowledge base as long as the necessary identifiers are URIs.

Feature 2: Matchings. GERBIL offers three types of matching between a gold standard and the results of annotation systems: a *strong entity matching* for URIs, as well as a *strong* and a *weak annotation matching* for entities.

Feature 3: Measures. Currently, GERBIL offers six measures subdivided into two groups: the micro- and the macro-group of precision, recall and f-measure. As shown in Figure 1(a), these results are displayed using interactive spider diagrams that allow the user to easily (1) get an overview of the performance of single tools and (2) compare tools.

Feature 4: Diagnostics. An important novel feature of our interface is that it displays the correlation between the features of data sets and the performance of tools (see Figure 1(b)). By these means, we ensure that developers can easily gain an fine-grained overview of the performance of tools and thus detect possible areas of improvement for future work.



(a) Example spider diagram of recent A2KB experiments with weak annotation matching.

(b) Spider diagram of correlations between annotation results and data set features.

Fig. 1: Spider diagrams generated by the GERBIL interface.

Feature 5: Annotators. Currently, GERBIL offers 9 entity annotation systems with a variety of features, capabilities and experiments out-of-the-box.

Feature 6: Data sets. The latest version of GERBIL offers 11 data sets. Thanks to the large number of formats, topics and features of the data sets, GERBIL allows carrying out diverse experiments.

Feature 7: Output. GERBIL's experimental output is represented as a table containing the results, as well as embedded JSON-LD[2] RDF data for the sake of archiving results. Moreover, GERBIL generates a permanent URI for each experimental result.

In this paper, we will give a detailed explanation of the different RDF data structures underlying GERBIL's architecture. We will explain the internal workflow of GERBIL

[2] `http://www.w3.org/TR/json-ld/`

and argue why it simplifies the implementation of further experiments, annotators, data sets, matchings and measures. We conclude by pointing at future work.

2 GERBIL's interfaces, dataflow, structure

2.1 Datastructures

GERBIL unifies the different formats used by existing datasets and annotators. To this end, GERBIL's interfaces are mainly based on the *NLP Interchange Format* (NIF). This is a RDF-based Linked Data serialization which provides several advantages such as interoperability by standardization or query-ability. The *NIF-standard* assigns each document an URI as starting point and generates another Linked Data resource per semantic entity. Each document is a resource of type `nif:Context` and its content is the literal of its `nif:isString` predicate. Every entity is an own resource with a newly generated URI pointing to the original document via the `nif:referenceContext` predicate. Additionally the begin (`nif:beginIndex`) and end position (`nif:endIndex`) as well as the disambiguated URI (`itsrdf:taIdentRef`) and the respective KB (`itsrdf:taSource`) are stored. NIF's paramount position amongst corpora serialisation formats is evident by the growing number of available datasets [6].[3]

GERBIL's main aim is to provide comprehensive, reproducible and publishable experiment results. Thus, GERBIL enforces the use of a machine-readable description for each experiment via JSON-LD[4] RDF data using the RDF DataCube vocabulary [4] next to a human-readable table presentation. The *RDF DataCube* vocabulary can be used to represent fine-grained multidimensional, statistical data which is compatible with the Linked SDMX [2] standard. GERBIL models each experiment as `qb:Dataset` containing `qb:Observations` for each individual run of a annotator on a dataset. Each observation features the `qb:Dimensions` experiment type, matching type, annotator, corpus, and time. The evaluation measures and an error count are expressed as `qb:Measures`.[5]

GERBIL relies on the DataID ontology [1] to represent further metadata as well as annotator and corpus information. Besides metadata properties like titles, descriptions and authors, the source files of the open datasets themselves are linked as `dcat:Distributions`, allowing direct access to the evaluation corpora. Furthermore, ODRL license specifications in RDF are linked via `dc:license`, potentially facilitating automatically adjusted processing of licensed data by NLP tools. Licenses are further specified via `dc:rights`, including citations of the relevant publications.[6] To describe annotators in a similar fashion, we extended DataID for services. The class `Service`, to be described with the same basic properties as dataset, was introduced.

[3] The prefix `nif` stands for `http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#` while `itsrdf` is short for `http://www.w3.org/2005/11/its/rdf#`.

[4] `http://www.w3.org/TR/json-ld/`

[5] `qb` is a prefix for for `http://purl.org/linked-data/cube#`.

[6] The prefix `dcat` stands for `http://www.w3.org/ns/dcat#` while `dc` is short for `http://purl.org/dc/elements/1.1/`.

To link an instance of a `Service` to its distribution the `datid:distribution` property was introduced as super property of `dcat:distribution`, i.e., the specific URI the service can be queried at. Furthermore, Services can have a number of `datid:Parameters` and `datid:Configurations`. Datasets can be linked via `datid:input` or `datid:output`.[7] An example JSON-LD for an archived experiment can be found below.

```
{
  "@graph" : [ {
    "@id" : "http://gerbil.aksw.org/gerbil/experiment?id=...#experiment_...",
    "@type" : [ "gerbil:Experiment", "qb:Dataset" ],
    "experimentType" : "gerbil:A2KB",
    "matching" : "gerbil:WeakAnnoMatch",
    "structure" : "gerbil:dsd",
    "label" : "Experiment 201503160001"
  }, {
    "@id" : "http://gerbil.aksw.org/gerbil/experiment?id=...#experiment_..._task_0",
    "@type" : "qb:Observation",
    "annotator" : "http://gerbil.aksw.org/gerbil/dataId/corpora/Babelfy",
    "dataset" : "http://gerbil.aksw.org/gerbil/dataId/annotators/ACE2004",
    "statusCode" : "-1",
    "timestamp" : "2015-03-16T12:31:52.469Z"
  } ],
  "@context" : {
    ...
  }
}
```

2.2 Workflow

Figure 2 shows the architecture of GERBIL with the data sets at the bottom, the annotators in the top and the user interface as well as user defined annotator and data set at the right. A GERBIL session starts at the configuration screen with which a user defines the experiment he is interested in. Each experiment is divided into tasks. A task comprises the evaluation of a single annotator using a single data set, is encapsulated into fault-tolerant classes and runs inside an own thread. Our fault-tolerance classes at two types of errors: (1) an annotator may return error codes for single documents, e.g., because of the missing ability to handle special characters. While other evaluation frameworks tend to cancel the experiments after an exception thrown by the annotator, GERBIL counts these smaller errors and reports them as part of the evaluation result. The second type of fault tolerance aims at (2) larger errors, e.g., the data set couldn't be loaded or the annotator is unreachable via its Web service. These run-time errors are handled by storing one of the predefined error codes inside the experiment database. Therewith, we ensure that the user gets instant feedback if some parts of the experiment couldn't be performed as expected.

During a task, the single documents of a data set are sent to the annotator. After finishing the last document, the responses are evaluated. Currently, the evaluation is focused on the quality, i.e., precision, recall, F1-score and error counts, but can be extended. Moreover, a runtime is also available [6]. For some experiment types, e.g., the entity-linking tasks, the evaluation needs additional information. GERBIL is able to search for `owl:sameAs` links to close the gap between data sets and annotators that

[7] `datid` is a prefix for for `http://dataid.dbpedia.org/ns/core#`.

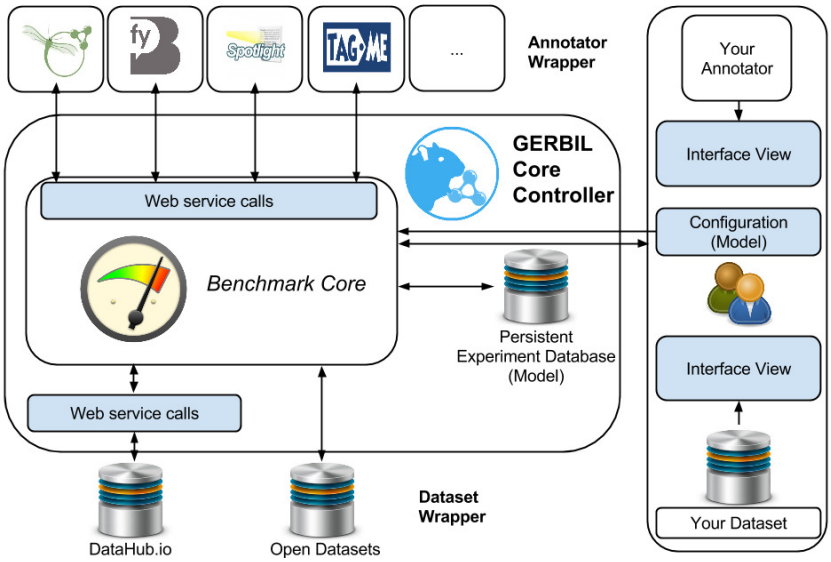


Fig. 2: Overview of GERBIL's abstract architecture. Interfaces to users and providers of data sets and annotators are marked in blue.

are based on different knowledge bases. Currently, this search is mainly based on the information inside the data set and retrieval of the entity mentioned by the annotator. The search could be extended by using local search indexes that contain mappings between well-known knowledge bases, e.g., DBpedia and Freebase. The results are currently written to an HSQL database[8].

2.3 Extensible Interfaces

The workflow of GERBIL is very general. An experiment has a certain experiment type, a matching, and a couple of datasets and annotators. Thus, it is easily possible to add new experiment types to GERBIL that are not part of the system, e.g., word sense disambiguation. One major advantage towards this form of extensibility is the usage of NIF for transferring the single documents. Since NIF is based on RDF the documents sent and received by the system as well as the datasets can be enriched with further information that can be used for the experiments. Thus, it will be easy to add a new experiment type even if the type needs information that cannot be expressed with NIF, e.g., the entity typing task defined in the Open Knowledge Extraction Challenge 2015[9]. An annotator that is able to identify the type of a new, unknown entity might add this type to its response. This information can't be understood directly by the response handling, but will be kept and made available to the evaluation component of GERBIL.

[8] http://hsqldb.org/
[9] http://2015.eswc-conferences.org/important-dates/call-OKEC

Thus, this type information will be available to evaluate the typing performance of an annotator.

3 Conclusion and Future Work

In this paper, we presented GERBIL, a platform for the evaluation, publishing and archiving of semantic entity annotation experiments. GERBIL extends the state-of-the-art benchmarks by dealing with data sets and annotators that link to different knowledge bases. Furthermore it offers extensible interfaces, reliable experiment descriptions as well as diagnostics and decision support. Our future work will comprise a better experiment task scheduling to achieve a higher efficiency. Another task is the improvement of the user interface towards a better intelligibility. Finally, we will devise a solution to ensure that GERBIL remains available to the community for the years to come.

Acknowledgments. Parts of this work were supported by the FP7 project GeoKnow (GA No. 318159) and the BMWi project SAKE (GA No. 01MD15006E).

References

1. M. Brümmer, C. Baron, I. Ermilov, M. Freudenberg, D. Kontokostas, and S. Hellmann. DataID: Towards semantically rich metadata for complex datasets. In *I-SEMANTICS*, 2014.
2. S. Capadisli, S. Auer, and A.-C. Ngonga Ngomo. Linked SDMX data. *Semantic Web Journal*, 2013.
3. M. Cornolti, P. Ferragina, and M. Ciaramita. A framework for benchmarking entity-annotation systems. In *22nd World Wide Web Conference*, 2013.
4. R. Cyganiak, D. Reynolds, and J. Tennison. The RDF Data Cube Vocabulary, 2014. http://www.w3.org/TR/vocab-data-cube/.
5. G. Rizzo, M. van Erp, and R. Troncy. Benchmarking the extraction and disambiguation of named entities on the semantic web. In *9th LREC*, 2014.
6. R. Usbeck, M. Röder, A.-C. Ngonga Ngomo, C. Baron, A. Both, M. Brümmer, D. Ceccarelli, M. Cornolti, D. Cherix, B. Eickmann, P. Ferragina, C. Lemke, A. Moro, R. Navigli, F. Piccinno, G. Rizzo, H. Sack, R. Speck, R. Troncy, J. Waitelonis, and L. Wesemann. GERBIL – General Entity Annotation Benchmark Framework. In *24th WWW conference*, 2015.