Entity-fishing: a DARIAH entity recognition and disambiguation service

Luca Foppiano, Laurent Romary

Entity-fishing: A DARIAH Entity Recognition and Disambiguation Service[1]

Luca Foppiano[2] and Laurent Romary[2]

Abstract

This paper presents an attempt to provide a generic named-entity recognition and disambiguation module (NERD) called entity-fishing as a stable online service that demonstrates the possible delivery of sustainable technical services within DARIAH, the European digital research infrastructure for the arts and humanities. Deployed as part of the national infrastructure Huma-Num in France, this service provides an efficient state-of-the-art implementation coupled with standardised interfaces allowing an easy deployment on a variety of potential digital humanities contexts. Initially developed in the context of the FP9 EU project CENDARI, the software was well received by the user community and continued to be further developed within the H2020 HIRMEOS project where several open access publishers have integrated the service to their collections of published monographs as a means to enhance retrieval and access. entity-fishing implements entity extraction as well as disambiguation against Wikipedia and Wikidata entries. The service is accessible through a REST API which allows easier and seamless integration, language independent and stable convention and a widely used service-oriented architecture (SOA) design. Input and output data are carried out over a query data model with a defined structure providing flexibility to support the processing of partially annotated text or the repartition of text over several queries. The interface implements a variety of functionalities, like language recognition, sentence segmentation and modules for accessing and looking up concepts in the knowledge base. The API itself integrates more advanced contextual parametrisation or ranked outputs, allowing for the resilient integration in various possible use cases. The entity-fishing API has been

[1] This research has received funding from the European Union's Horizon 2020 (H2020) research and innovation programme under grant agreement No 731102 (HIRMEOS).

[2] ALMAnaCH, Inria Paris, Inria

used as a concrete use case to draft the experimental stand-off proposal, which has been submitted for integration into the TEI guidelines. The representation is also compliant with the Web Annotation Data Model (WADM). In this paper we aim at describing the functionalities of the service as a reference contribution to the subject of web-based NERD services. In this paper, we detail the workflow from input to output and unpack each building box in the processing flow. Besides, with a more academic approach, we provide a transversal schema of the different components taking into account non-functional requirements in order to facilitate the discovery of bottlenecks, hotspots and weaknesses. We also describe the underlying knowledge base, which is set up on the basis of Wikipedia and Wikidata content. We conclude the paper by presenting our solution for the service deployment: how and which the resources where allocated. The service has been in production since Q3 of 2017, and extensively used by the H2020 HIRMEOS partners during the integration with the publishing platforms.

Keywords: named entity recognition and disambiguation, infrastructure, conditional random fields, service

1 Introduction

This paper describes an attempt to provide a generic named-entity recognition and disambiguation module (NERD) called entity-fishing as a stable online service. The work intends to demonstrate the possible delivery of sustainable technical services as part of the development of research infrastructures for the humanities in Europe. In particular, the developments described here contribute not only to DARIAH, the European Digital Research Infrastructure for the Arts and Humanities[3], but also to OPERAS, the European research infrastructure for the development of open scholarly communication in the social sciences and humanities[4]. Deployed as part of the French national infrastructure Huma-Num[5], the service provides an efficient state-of-the-art

[3] Accessed July 7, 2020, https://www.dariah.eu/

[4] Accessed July 7, 2020, https://operas.hypotheses.org/.

[5] Accessed July 7, 2020, https://www.huma-num.fr/.

implementation coupled with standardized interfaces allowing easy deployment in a variety of potential digital humanities contexts. In this paper we focus on the integration of entity-fishing within the European HIRMEOS project[6], where several open-access publishers have integrated the service into their collections of published monographs as a means to enhance retrieval and access. In the following sections, we give a quick overview of the accessibility and sustainability issues we want to address through our experiment, explain the general context of the HIRMEOS project, and then provide a comprehensive description of various facets of entity-fishing: its architecture, interfaces, and deployment.

2 Providing Access in Service of Sustainability

As already alluded to by Romary and Edmond (2017), the sustainability of digital services is strongly related to reusability, in that the actual deployment of tools and services in workable research scenarios and environments is key to ensuring their further maintenance, updating, and long-term availability. This relation to users is also tied to the capacity of the service to facilitate the research activity without putting constraints on the way the research itself shall be carried out, following the analysis by Edwards (2003) in the specific context of research infrastructures.

In the context of online services such as *entity-fishing*, which aims at enriching digital documents by means of stand-off annotations, we have listened to the users' needs and therefore have driven our contribution to sustainability along the following axes:

The definition of an **open and flexible architecture** based upon open-source components, so that the software maintenance can be partially or totally transferred to a third party at any time

The provision of **standardized interfaces** which cover various use cases ranging from the direct provision of standardized annotations (e.g., compliant with the TEI

[6] High Integration of Research Monographs in the European Open Science infrastructure, accessed July 7, 2020, https://www.hirmeos.eu.

Guidelines: see TEI Consortium 2020) to the integration of the service in more complex technical environments

The deployment of the service as a **dedicated professional environment** within the public sphere, ensuring a reliable and scalable usage right from the beginning of the integration phase in concrete use cases

Finally, and most importantly, the anchorage on **real use cases** reflecting situations where the service provides a distinct added value to the existing digital resources made available to humanities scholars

In this paper, we show how we have addressed these different aspects, but we would first like to provide some background on the genesis of the *entity-fishing* project and how it has reached the stage of becoming a generic online service for DARIAH and OPERAS.

3 From Cendari-NERD to *entity-fishing* in HIRMEOS

The development of *entity-fishing* began in the context of the EU FP7 Cendari project from 2013 to 2016 (Lopez, Meyer, and Romary 2014), which aimed at setting up a digital research environment for historians specializing in the medieval and First World War periods that would facilitate their access to archival content (Vanden Daelen et al. 2015) and enable them to acquire information about the various assets, or entities, involved in their research scenarios. At an early stage in the project, it was determined that the provision of automatically extracted entities from the various sources (primary or secondary) that the historians are working with would boost the selection of appropriate material for the research at hand. Initially deployed in the technical framework of Inria[7], the *entity-fishing* service gained considerable interest from a variety of users, not just historians, who continued to use it on a regular basis long after the project had ended, which in turn put pressure on us to further maintain and enhance it.

[7] Institut national de recherche en sciences et technologies du numérique, accessed July 24, 2020, https://www.inria.fr/en.

We thus took the opportunity of the EU Horizon 2020 (H2020) HIRMEOS[8] project to further consolidate and expand the service. Indeed, HIRMEOS addresses the peculiarities of academic monographs as a specific support for scientific communication in the social sciences and humanities. It aims to prototype innovative services for monographs by providing additional data, links to, and interactions with the documents, at the same time paving the way to new potential tools for research assessment, which is still a major challenge in the humanities and social sciences.
In particular, HIRMEOS sets up a common layer of services on top of several existing e-publishing platforms for open access monographs. The goal of the *entity extraction* task was to deploy the service and process open access monographs provided by the HIRMEOS partners. The documents available were the following:

4,000 books in English and French from Open Edition Books[9]
2,000 titles in English and German from OAPEN[10]
162 books in English from Ubiquity Press[11]
765 books (606 in German, 159 in English) from the University of Göttingen Press[12]

The introduction of *entity-fishing* has undergone different levels of integration. The majority of the participating publishers provided additional features in their user interface, using the data generated by *entity-fishing*, for example, as search facets for persons and locations to help users narrow down their searches and obtain more precise results.

[8] High Integration of Research Monographs in the European Open Science infrastructure, accessed July 24, 2020, http://www.hirmeos.eu.

[9] Accessed July 10, 2020, https://books.openedition.org/.

[10] OAPEN (Open Access Publishing in European Networks), accessed July 10, 2020, http://www.oapen.org/.

[11] Accessed July 10, 2020, https://www.ubiquitypress.com/.

[12] Universitätsverlag Göttingen, accessed July 10, 2020, https://www.univerlag.uni-goettingen.de/.

4 Entity Extraction

Identifying entities is a central task in the analysis of secondary scholarly literature (Brando, Frontini, and Ganascia 2016). Such entities may range from simple key terms to very specific scientific, nomenclature-based expressions (chemical formulas, astronomical objects, expressions of quantities, etc.). It also covers regular named entities such as person-names or locations, which correspond to core tasks in the social sciences and humanities (Smith and Crane 2001).

From the applicability point of view, entity extraction is a task that is also suitable for small quantities of data at document level: for instance, in the case of the Amazon X-Ray functionality of the Kindle (Wright 2012), proposing a list of people appearing in a book. Extracting varieties of data allows the system to answer questions related to the document itself before it has been read.

On a larger scale, with an increasing number of documents, the resulting graph of interlinked connections allows the computing of aggregated information such as trends, semantic search, or document similarity.

Authors making the choice of open-access publishing may also have a particular interest in having their work more frequently discovered, read, used, and reused to get credit and recognition. Therefore, it is the duty of the corresponding infrastructure to provide the means to accomplish these objectives. HIRMEOS is a great opportunity to improve the current open access technical infrastructure at the European scale.
In following subsections we describe the *entity-fishing* service. Sections 4.1 and 4.2 provide background information; then we describe the architecture and data flow in section 4.3. Section 4.4 presents details on the standard interfaces, and sections 4.4 and 4.5 conclude with the knowledge base (KB) organization and the external data source.

4.1 Disambiguation: From Mentions to Entities

In this paper, we define *mention* as a textual segment, one or a combination of words, that can be identified in the text. Mentions usually are strongly dependent on the domain: for example, analyzing the same text from biology and chemistry perspectives would output different mentions. We define *entity* (following Wikipedia) as "something that exists as itself, as a subject or as an object, actually or potentially, concretely or

abstractly, physically or not."[13] Therefore, *linking* is the task of finding a KB reference that a particular mention may refer to within the current context. Finally, a *mention* linked to a concept in the KB is an *entity*.

We have defined entities and mentions separately because their identification is decoupled in two different subsequent processes. The *mention extraction* is performed by means of generic parsers (known as NER, Named Entity Recognition) or by plugging in specialized parsers when dealing, for instance, with dates, ancient names, or chemical formulas.

The *entity disambiguation* task (also called *entity linking*, *named entity disambiguation*, *named entity recognition and disambiguation*) consists of determining the actual identity of the entity which is referred to by expressions appearing in a document. There is a long-standing body of research that aims to improve efficiency and completeness of the extraction of relevant information and senses depending on the context provided by the text, as well as the reference background provided by large-scale entity databases such as Wikipedia (Cucerzan 2007).

In fact, *entity disambiguation* requires a knowledge base that contains all the entities to which each mention may be linked. With its open license, Wikipedia has become the reference knowledge base (Milne, Witten, and Nichols 2007) for such a task. Ratinov et al. (2011) formally define *Wikification* as the task of identifying and linking expressions in text to their referent Wikipedia pages. In order to be more generic, mentions can be recognized using several techniques (based on Machine Learning [ML], lexicons, or rules), extended to any kind of expression beyond the narrow notion of "named entities." However, since Wikipedia—although comprehensive—has some sort of physical limitations, it could happen that certain entities identified in a text are not found at all in the knowledge base. More details about Wikipedia and Wikidata will be discussed in section 4.5.

To illustrate this phenomenon, we can take the following text as an example:

President Obama is living in Washington

There are two mentions, *President Obama* and *Washington*, in the sentence. If we consider our approach based on Wikipedia, the *entity-fishing* knowledge base terms lookup will say that there are respectively 10 possible candidates for *Obama* and 715 for *Washington*. A

[13] DBpedia page, accessed July 24, 2020, http://dbpedia.org/page/Entity.

correct disambiguation will be the link of the mention *Washington* to the entity titled *Washington D.C.*, described as "Washington, D.C., formally the District of Columbia and commonly referred to as Washington or D.C., is the capital of the United States of America"[14]. and identified by Wikipedia page ID *108956* and Wikidata ID *Q61*. Moreover, the mention *Obama* should be correctly disambiguated to *Barack Obama* with Wikipedia page ID *534366* and Wikidata ID *Q76*.

4.2 *Entity-fishing*

Entity-fishing is a service implementing the task of entity recognition and disambiguation using both Wikipedia and Wikidata. It is designed to be domain agnostic, thus giving the flexibility to be upgraded to support disambiguation of specialized entities with minimal effort. It currently supports five languages: English, French, German, Italian, and Spanish.

Entity-fishing can process raw text with optional pre-identified mentions or entities which are used to help resolve ambiguities. It supports search queries—short text with very minimal context—and PDFs. PDF support is provided by the GROBID library (Lopez 2009).

The service provides a web interface (fig. 1) and a REST API (4.4) for third-party service integration.

[14] https://en.wikipedia.org/wiki/Washington,_D.C., accessed August 31, 2020.

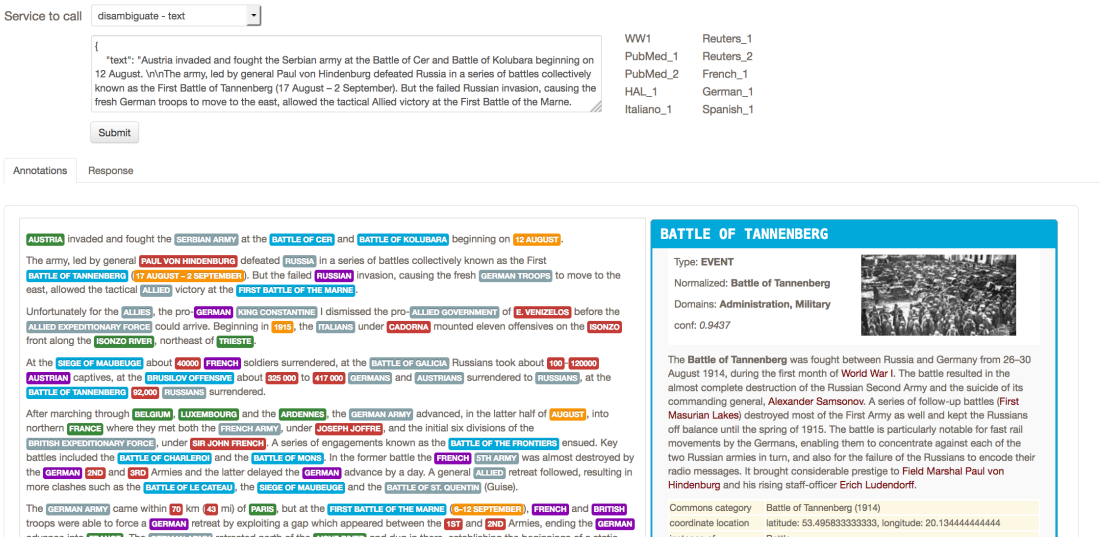

Figure 1. Example of text about World War I as analyzed in the *entity-fishing* web interface

4.3 Architecture

In this section we describe the system architecture from two orthogonal viewpoints: first, we show how the system works from a data-flow perspective, to understand the functional aspects of the system from input to output. Second, we focus on non-functional requirements in order to identify bottlenecks and critical components, thus allowing a correct definition of requirements.

4.3.1 Data-driven Architecture

We describe here in detail the main steps occurring between input and output as outlined in figure 2, ignoring how input and output are actually represented (sec. 4.4.1).

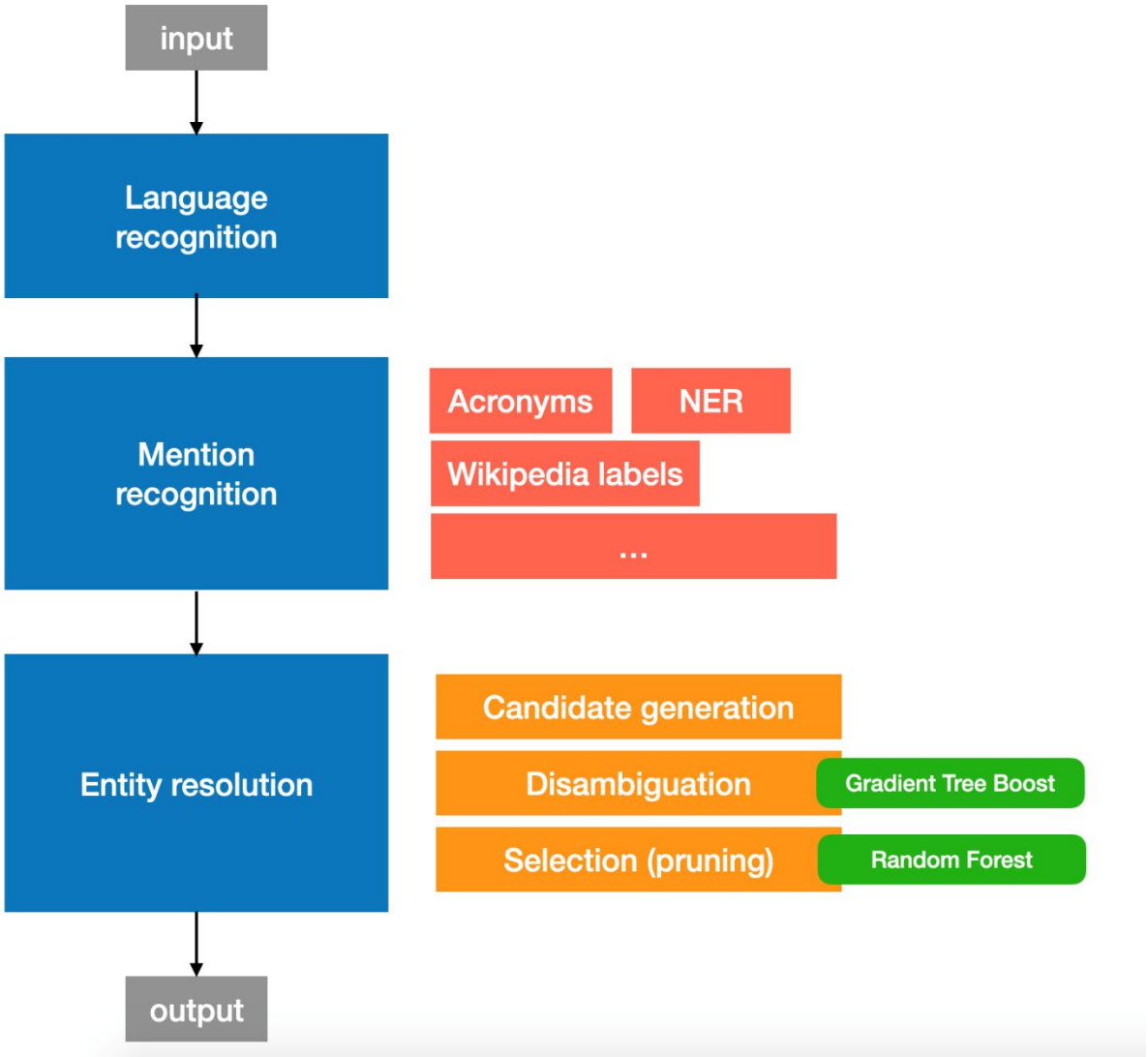

Figure 2. The data-driven architecture of *entity-fishing*

The service can be divided into three steps: language identification, mention recognition, and entity resolution.

4.3.1.1 Language Identification

Language is an important variable in the *entity-fishing* process: it is used to select the appropriate utilities for processing text, such as the tokenizer and the sentence

segmenter, and, most importantly, to select the specific Wikipedia (as of July 2020 there are 300 active Wikipedias[15]) from the knowledge base (sec. 4.5). This step is ignored when the information is provided by the user.

4.3.1.2 Mention Recognition

This component is responsible for extracting mentions (4.1) from the input. The idea is to have a generic set of recognizers bundled with the system and offer users the option to extend to specific domains by plugging in additional ones.

Entity-fishing ships three traditional mention extractors:

- Named Entity Recognition
  The term Named Entity was coined during the MUC-6 evaluation campaign (Nadeau and Sekine 2007) and was specifically referring to entity name expressions (e.g., persons, locations, and organizations) and numerical expressions (e.g., measures). The Named Entity Recognition in *entity-fishing* uses GROBID-NER[16] (initially written by Patrice Lopez; we have contributed to improving it), a library for processing text and extracting Named Entities classified into 27 classes[17] using a Conditional Random Field (CRF) statistical model. The extracted mentions are not limited to those observed in the training data thanks to the generalization properties of the ML processes; however, they are not guaranteed to be systematically resolved in Wikipedia/Wikidata.
- Wikipedia lookup
  This method is complementary to the ML NER approach. It offers more stable results, which are constrained by Wikipedia coverage: entities that are not in Wikipedia cannot be found. The lookup attempts to find all mentions that correspond to either a title or an anchor (and variants thereof) in Wikipedia using an *n*-gram-based matching approach (with $n = 6$).
- Acronyms extraction

---

[15] List of Wikipedias, accessed July 24, 2020, https://en.wikipedia.org/wiki/List_of_Wikipedias.

[16] GitHub repository, accessed July 8, 2020, http://github.com/kermitt2/grobid-ner.

[17] "Classes and Senses," GROBID NER User Manual, accessed July 8, 2020, http://grobid-ner.readthedocs.io/en/latest/class-and-senses/.

An acronym is a word (usually a name) composed as an abbreviation from the initial components in a phrase, usually from individual letters (as in NATO or laser) but sometimes also from syllables (as in Benelux). Acronyms and initialisms (abbreviations formed like acronyms but pronounced letter-by-letter) are widely used in research articles in order to optimize space and to communicate concepts and methods more effectively (e.g., humoristic acronyms are usually easier to remember). The lexical structure of an acronym or initialism can be approximated as a mapping pair (acronym/initialism, base). For example, *DIY* can be represented as (DIY, Do It Yourself). In scientific articles such mappings are presented once at the beginning of the paper or in the abstract. *Entity-fishing* treats the acronyms and initialisms as mentions and uses their *base* for disambiguation label. The resolved entity is then further propagated in the text for each occurrence of the acronym.

The result from this step is a list of *Mention* objects, containing raw value from the original text, positions (offsets or coordinates), and NER type (within the 27 classes extracted from GROBID-NER (4.3.1.2)).

4.3.1.3 Entity Resolution

Entity resolution is the process of linking each mention to the authority records in Wikipedia and Wikidata through their identifiers.

The resolution process consists of three further phases (fig. 2):

- Candidate generation
  In this phase, each mention is linked to a list of concepts (possible candidates for the disambiguation) matching—entirely or with some variation—the original raw name.
- Candidate ranking
  Each candidate is assigned a confidence score calculated as regression probability from an ML model based on gradient tree boosting using features from local (related to the concept itself) and remote (related to the concept and its relationship with other concepts) information. The features used in our candidate ranking process are:
  - Milne & Witten relatedness measure (Witten and Milne 2008) between the candidate and the context (list of mentions appearing in the text)

  - Entity centroid score calculated using entity and word embedding, combining the entity representation and the context within which the entity occurs (using a window of 10 words)
  - Prior probability, or commonness: the probability that mentions in Wikipedia correspond to an effective link to the candidate. Lower scores are assigned to more ambiguous concepts
  - Context quality: an evaluation of how the concepts composing the context are related to each other, with higher scores assigned for a larger number of related concepts.
- Candidate selector

  The candidate list is pruned comparing a calculated *selection score*, with minimal values selected manually, for each language. The selection score is calculated as the output of an ML regression model computing the following features:
  - Ranker score, the score calculated by the candidate ranker
  - Link probability, which returns the probability that the label associated with the candidate is used as a link in Wikipedia (calculated as the number of articles that contain links with this label used as an anchor, divided by the number of articles that mention this label)
  - Prior probability or commonness: the probability that mentions in Wikipedia correspond to an effective link to the candidate
  - Milne & Witten relatedness measure (Witten and Milne 2008)
  - TF-IDF, Term Frequency–Inverse Document Frequency (Salton and Michael 1983), which reflects how important a term (here, the mention) is to a document in a collection or corpus
  - Named entity, a Boolean value set to true if the mention to be disambiguated is also a named entity (meaning that it has been classified as such in the mention recognition process)
  - Generalized Sørensen–Dice coefficient of a term (candidate normalized name) given global Wikipedia frequencies used to capture lexical cohesion

The final output consists of a list of entities. In some cases is possible that no entry in the knowledge base is retrieved: for example, a PERSON mentioned in the text only by first name. Similarly, named entities of class MEASURE are not disambiguated at all.

4.3.2 Service Component Architecture

Most of the work carried out in the HIRMEOS project aims to measure and improve the robustness and scalability of the service. In particular we envisioned, right from the onset, that the service would reach out to a large group of potential users.

We present three layers into which the service can be decomposed, each of them being characterized by different requirements and constraints: the web interface, the engine, and the data storage (fig. 3).

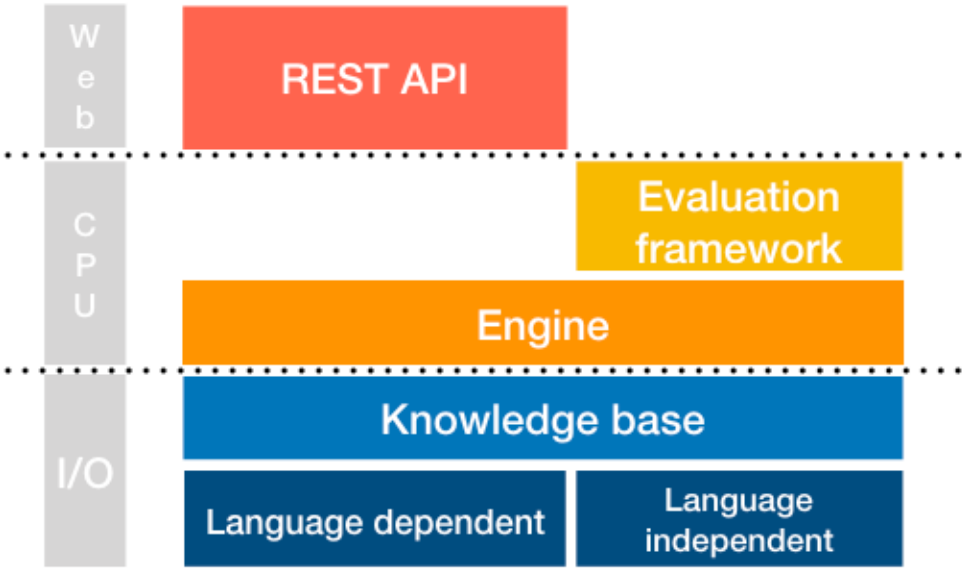

Figure 3. Architecture of *entity-fishing* architecture

4.3.2.1 Web Interface

The web interface provides a REST API presenting services as HTTP entry points (sec. 4.4.1). The main responsibility of the service is to understand, validate, and process requests (e.g., avoiding malformed or incomplete queries and verifying the correctness

of the input data) and to yield, when applicable, the right error code and message (e.g., returning 406 when the language is not provided, 400 if the query is malformed). This component does not pose any performance threat.

4.3.2.2 Data Storage

The data storage layer is responsible for maintaining the large amount of data which cannot be kept in memory. The service is handling the entire contents of the Wikipedias (see sec. 4.6) for our different languages (each consisting of several gigabytes of raw data) and precomputed data (such as the number of links per article source and destination, or the number of documents).

During normal service operations, *entity-fishing* reads only from data storage, which "simplifies" the requirements (e.g., no risk of dirty reads and/or performance issues). We evaluated different storage technologies: NOSQL Databases such as MongoDB were discarded because of their complexity. We selected a light key-value database with memory-mapped files: Lightning Memory-Mapped Database (LMDB). Developed by Symas [18], it is fast storage for the OpenLDAP project that fully supports ACID semantics, concurrent multithread read/write access, transactions, and zero-copy lookup and interaction, and is available with bindings for several languages (Java, Python, etc.). *Entity-fishing* uses a Java library called lmdbjndi [19] to implement it. The data storage layer currently consists of one database with the language-independent information (Wikidata metamodel) and five language-specific databases containing Wikipedia information, each of which holds 23 collections, for a total of about 80 GB of required disk space. This is indeed the most critical part of the system.

4.3.2.3 Engine

The engine is the main component of the application, as it orchestrates the various steps (mention recognition, entity resolution, etc.) originating from a query (and possibly a file) provided as input. The engine interacts heavily with the data storage layer to compute the features and retrieve all the information.

[18] Accessed July 8, 2020, https://symas.com/lmdb/.

[19] LMDB JNI GitHub repository, accessed July 8, 2020, https://github.com/deephacks/lmdbjni.

4.4 Standard Interface

4.4.1 REST API

The REST API provides a standard entry point interface. The REST protocol is the standard means of communication for service and microservice integration. The API has three main access points: the disambiguation process, the KB access, and a set of secondary utilities. The disambiguation entry point takes as input a JSON query like:

```
{
  "text": "The text to be processed.",
  "shortText": "term1 term2 ...",
  "language": {
    "lang": "en"
  },
  "entities": [],
  "mentions": ["ner","wikipedia"],
  "nbest": 0,
  "sentence": false,
  "customisation": "generic",
  "processSentence": []
}
```

Note that the first two elements in the input are mandatory and mutually exclusive. While *text* is used for long text (paragraphs), *short text* is used for search query disambiguation (fewer than 5 words), which requires a different approach because of the small amount of information available in a query.

Everything else is optional:

- `"language"`: the language provided by the user, which otherwise will be automatically recognized
- `"mentions"`: indicates the modules to be applied to the text, as described in section 4.3.1. The value “ner” corresponds to the Named Entities Recognition module, while “Wikipedia” indicates the lookup on the wikipedia knowledge base. Acronyms are always applied.

- `"nbest"`: return the *n* best disambiguated results (by default only the first one)
- `"entities"`: represents a list of entities already known by the user
- `"sentence"`: a Boolean value which, when set, results in segmenting the text in sentences (value `true` here will return an additional property called `"sentences"` containing a list of offsets identifying each sentence)
- `"processSentence"`: a list index (referring to the `"sentences"` properties: see above) of sentences to be processed (for long texts, it is good practice to proceed paragraph by paragraph)

The service returns a structure based on the query, with the output results. See example (the listing has been simplified):

```
{
    "text": "Austria invaded and fought the Serbian
    army at the Battle of Cer and the Battle of Kolubara
    beginning on 12 August.",
    "language": {
        "lang": "en"
    },
    "entities": [
         {
            "rawName": "Serbian Army",
            "offsetStart": 31,
            "offsetEnd": 43,
            "nerd_selection_score": 0.6148,
            "wikipediaExternalRef": 10072531,
            "wikidataId": "Q1209256"
            [...]
         },
         {
            "rawName": "Austria",
            "offsetStart": 0,
            "offsetEnd": 7,
            "type": "LOCATION"
```

```
            [...]
         }
    ]
}
```

This approach simplifies iterative processing workflows, which are required for processing long texts over several queries.

The KB REST interface allows full access to:

- concepts, to fetch a single concept using their Wikidata or Wikipedia identifier
- term lookup, verifying whether a label can be found in the knowledge base and how ambiguous it is

The secondary utilities are:

- text segmentation
- language recognition

A detailed description of these functionalities, which is beyond the scope of this paper, can be found in the official *entity-fishing* documentation[20].

4.4.2 TEI Representation

The work carried out in the HIRMEOS project also provided the opportunity to specify a TEI-compliant output for the *entity-fishing* service that would be easy to integrate within the ongoing stand-off proposal under discussion within the TEI Council (Banski et al. 2016). This proposal is based upon the concept of embedded stand-off annotation, where a `<standOff>` element gathers all the annotations related to the corresponding `<text>` element of a TEI document and is positioned between the `<teiHeader>` and the `<text>` elements, as illustrated below.

```
<TEI>
   <teiHeader>...</teiHeader>
```

[20] "entity-fishing – Entity Recognition and Disambiguation," accessed July 10, 2020, http://nerd.readthedocs.io/.

```
   <standOff>
      <teiHeader>...</teiHeader>
      <listAnnotation type="individuals"> ... </listAnnotation>
      <listAnnotation type="places">...</listAnnotation>
      <listAnnotation type="organizations">...</listAnnotation>
      <listAnnotation type="dateTimes"/>
      <listAnnotation type="events"/> ...
   </standOff>
   <text>...</text>
</TEI>
```

We can see how it is possible, for instance, to organize annotations in various groups: in our case, we have gathered entities by types.

In this framework, elementary annotations are structured as `<annotationBlock>`s containing three components:

- a `<span>` element that introduces a character interval within a numbered component in the text (mainly `<p>` elements)
- a specific element describing the entity (e.g., `<person>`) that has been recognized
- an `<interp>` element that links the interval with the entity

This representation is intended to be compliant with the target/body/annotation triptych of the Web Annotation Data Model[21]. It is illustrated below for the annotation of a person entity.

```
<annotationBlock>
 <person xml:id="_6117323d2cabbc17d44c2b44587f682c">
  <persName type="rawName">John Smith</persName>
 </person>
 <interp ana="#_6117323d2cabbc17d44c2b44587f682c"
  inst="#_b02f28110aa52495b3ec386d171bc20f"/>
 <span from="#string-range(//p[@xml:id='
```

[21] W3C Recommendation 23 February 2017, accessed July 8, 2020, https://www.w3.org/TR/annotation-model/.

```
 _899607cd21ce83fb3a8e35652ace2479'],259,269)"
  xml:id="_b02f28110aa52495b3ec386d171bc20f"/>
</annotationBlock>
```

The difficulty associated with the generation of TEI-based stand-off annotations is twofold: a) the risk of a deviation between the textual content and the character offsets which have been computed and b) the necessity to number the various elements that serve as anchors for the annotations (paragraphs, etc.). This difficulty is exactly what is behind the notion of embedded stand-off, which relies on the assumption that annotations and the corresponding reference version of the textual content represented in TEI are delivered together as one single coherent document instance. This is the way our service has been implemented, so that the output will be the one and only version of record for any further processing within the client that called the *entity-fishing* service.

4.5 External Data Sources

In this section we set out to provide a definition of our concept of knowledge base and then discuss in greater depth the origin of the various data sources we used. A knowledge base is defined as the set of information describing a certain domain of interest. It usually covers a specific area of knowledge (e.g., chemistry, biology, or astronomy). On the other hand, there are also generic knowledge bases such as Wikidata, DBPedia, or Freebase which are not bound to any specific domain. *Entity-fishing*, being a generic tool, essentially anchors its knowledge base upon Wikipedia and Wikidata. It is constructed by an offline process taking the dumps from Wikipedia and Wikidata as input, and aggregating the data in an appropriate structure and format which can be used efficiently by the service.

4.5.1 Wikipedia

Wikipedia is a multilingual, web-based, free-content encyclopedia, and is currently the largest and most popular general reference work on the internet. Owned and supported by the Wikimedia Foundation, a non-profit organization supported by donations, it was launched on January 15, 2001, by Jimmy Wales and Larry Sanger and initially only supported

the English language[22]. Other languages were quickly developed in the months following the launch. With over 6 million articles as of this writing, the English Wikipedia is the largest of the Wikipedia encyclopedias[23]. Wikipedia has reached a high level of completeness and popularity, with more than 50 million articles in over 300 different languages. Statistics[24] from the Wikimedia Foundation show popularity in the order of billion page-view and million of unique visitors each month. Its level of coverage and reliability has made Wikipedia a reference database for many information extraction processes.

4.5.2 Wikidata

As stated on its web site, "Wikidata is a free, collaborative, multilingual, secondary database, collecting structured data to provide support for Wikipedia, Wikimedia Commons, the other wikis of the Wikimedia movement, and to anyone in the world."[25]

Wikidata is a collaboratively edited knowledge base hosted by the Wikimedia Foundation (Vrandečić 2012). It is intended to provide a common source of data which can be used by Wikimedia projects such as Wikipedia, and by anyone else, under a public domain license (CC-0). Wikidata has become widely used: for example, Google decided to migrate its knowledge base Freebase into Wikipedia (Pellissier Tanon et al. 2016). Wikidata is the largest collaborative database in the world, administered by only six staff members with more than 18,000 human collaborators and several (semi-)automatic bots (Steiner 2014). In addition, it has increasingly become a reference knowledge base for many scientific disciplines: in 2014 Wikimedia announced the storage of the whole human genome[26].

---

[22] https://en.wikipedia.org/wiki/Wikipedia:About

[23] Wikipedia language index, https://en.wikipedia.org/wiki/Special:SiteMatrix

[24] https://stats.wikimedia.org

[25] Wikidata:Introduction, accessed July 9, 2020, https://www.wikidata.org/wiki/Wikidata:Introduction.

[26] Jens Ohlig,"Establishing Wikidata as the Central Hub for Linked Open Life Science Data" (blog post), October 22, 2014, https://blog.wikimedia.de/2014/10/22/establishing-wikidata-as-the-central-hub-for-linked-open-life-science-data/.

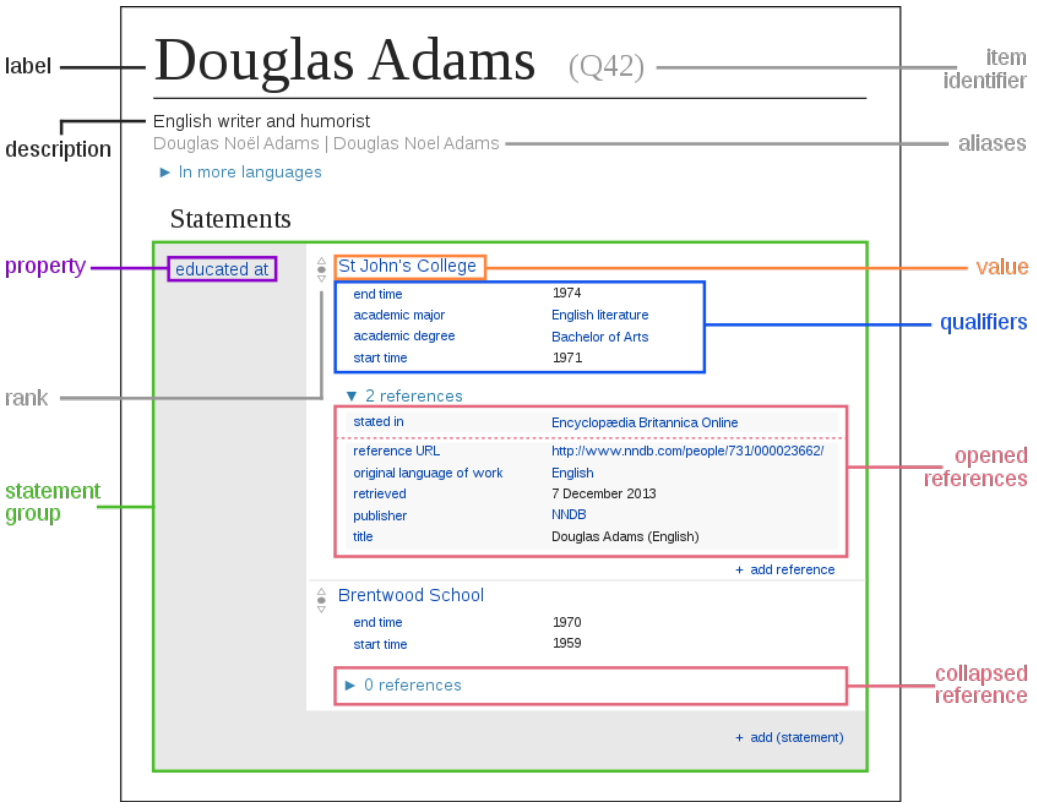

Figure 4. Data model in Wikidata

Wikidata provides a generic representation model where the elementary unit is a specific entity, identified by a unique ID starting with Q: for example, Douglas Adams has the ID Q42 (fig. 4). Each entity is modeled according to various properties: labels, aliases, translations, and their characteristics identified as statements. Statements are key-value information structures: the key is called *property*, uniquely identified by IDs starting with P (and reused throughout the whole database), such as *educated at* (P69).

Wikidata's motto is "verifiability, not truth," which means that each statement is supported by optional "references" providing verifiable sources of information. For example, the population of Berlin can be different depending on the source and other variables (such as the date of the measurement) (fig. 5). This approach is different from the Semantic Web, which assumes there is no contradictory information (axiomatic logic), and is actually much more powerful and appropriate given the way knowledge and science is produced.

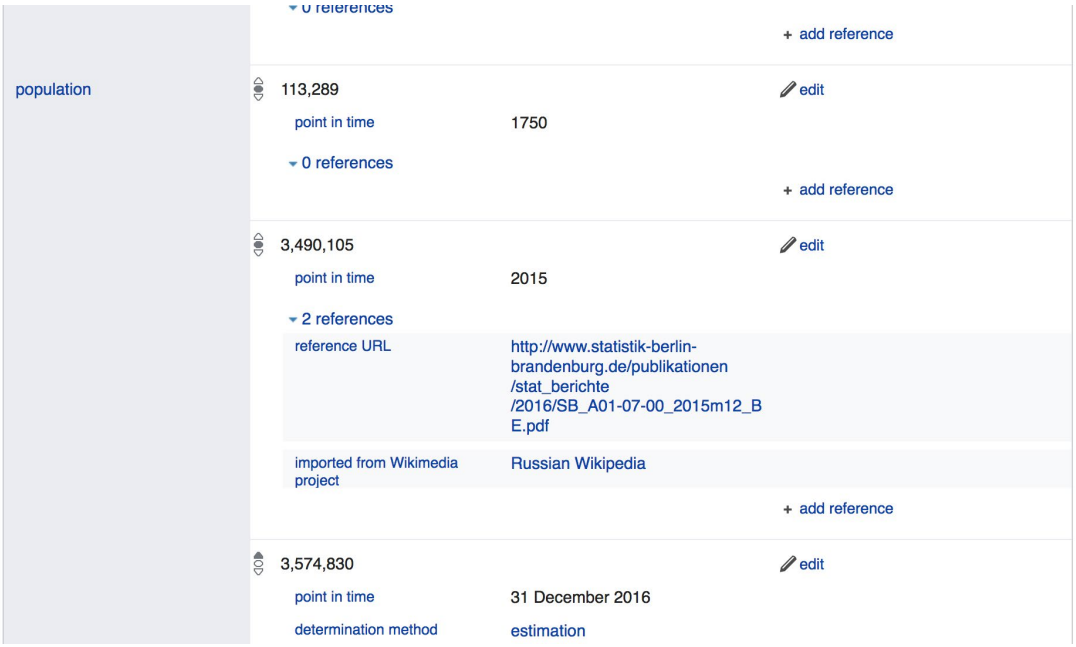

Figure 5. Example of how the references provide sources to support the statement

More information about the Wikidata metamodel can be found in the official documentation[27].

4.5.3 Motivation and Rationale

The decision to use Wikipedia and Wikidata as data sources is justified by the fact that both are generic and provide a basic, stable, and fairly complete knowledge set that can be systematically enriched by specialist domain data. Compared with other sources of information, they are the most complete available to use, reuse, and redistribute. Wikipedia is released under a CC-BY (attribution) license while Wikidata has chosen CC-0 (full copyright waiver). We would like to stress here the importance of the licensing choice in the long term, particularly when dealing with scientific knowledge.

Many people have shown concern about the fact that Wikipedia, being a collaborative source of information, could be biased by the contributors' opinions. This concern has a real

27 "Wikidata:Introduction," accessed July 10, 2020, https://www.wikidata.org/wiki/Wikidata:Introduction.

foundation, without any relevant impact on the results from the *entity-fishing* process. The way *entity-fishing* exploits Wikipedia relies on the graph network of concepts rather than the deep meanings of the articles themselves.

Finally, the use of externally managed data sources is strongly motivated and well advocated in software engineering best practices: maintainability and independent management, respectively. First, the amount of information managed is too big to be handled internally by the *entity-fishing* project itself. Second, having an independent body (the Wikimedia Foundation) administering the sources assures the relevant competences will be available for management, engineering, and content.

4.5.3.1 Multi-language Support

As of July 2020, the system supports English, Italian, French, German, and Spanish. The ability to support a language is strictly related to the number of articles available in the localized Wikipedia[28]. Languages with less than one million Wikipedia articles are not guaranteed provide consistent results.

4.6 Knowledge Base Organization and Access

4.6.1 Basic Organization

In this section, we examine how the data have been organized and integrated into the knowledge base.

Earlier in this paper (sec. 4.3.2 and fig. 3) we mentioned that the knowledge base is divided into two main areas corresponding to *language dependent* and *language independent* information, respectively. The language independent part, corresponding to the generic data provided by Wikidata, contains metadata across all languages and is accessed through the

[28] "List of Wikipedias," subsection "1 000 000+ articles," accessed July 9, 2020, https://meta.wikimedia.org/wiki/List_of_Wikipedias#1_000_000+_articles.

`UpperKnowledgeBase` object. Language dependent information, representing a Wikipedia in a specific language, is offered via the `LowerKnowledgeBase` object (fig. 6).

The data are accessed by means of the language-independent component (`UpperKnowledgeBase`) to get the general concept and further drawing on available specific language resources through the `LowerKnowledgeBase`. This component provides convenient access to all of the Wikipedia content through the programmatic API.

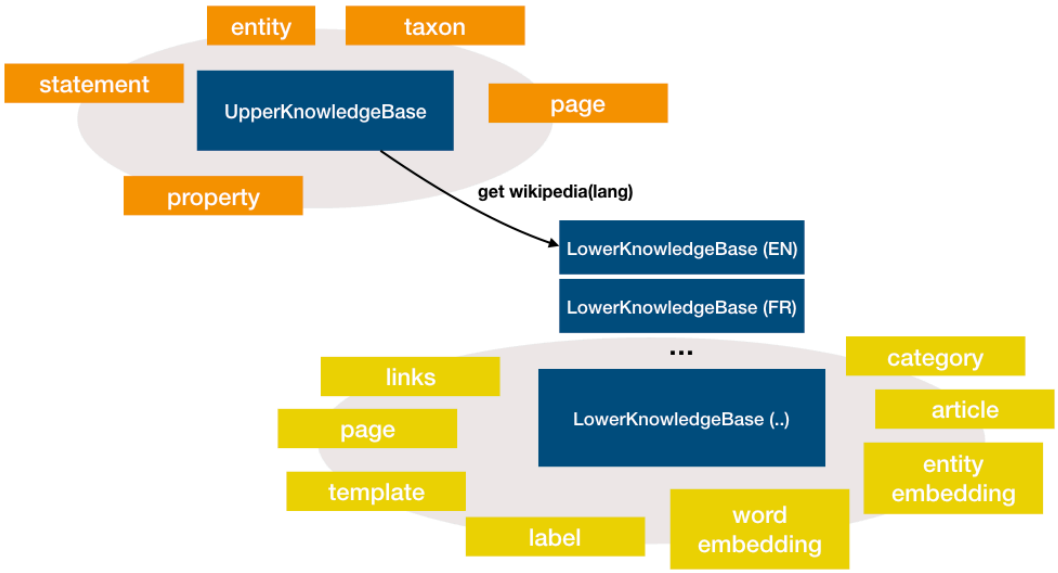

Figure 6. Schema of the knowledge base

4.6.2 API

Efficient access to the knowledge base is a crucial aspect of the disambiguation process as well as for additional developments and complementary integrations. Therefore, the service provides two means of access: through a Java API or a simpler REST API.

The REST API can be used to fetch the JSON representation of a concept, when provided with either a Wikipedia page ID, Wikidata item (Q123), or Wikidata property (P356, doi). The Java API provides direct access to the Java data model, which is not limited to concept retrieval but also covers precalculated information such as the number of entities, lookups, anchors, or disambiguation pages.

5 *Entity-fishing* as a Service

In this section we describe the process and the outcome of providing *entity-fishing* as a standalone service within the DARIAH infrastructure, from the technical requirements to the implemented solution. The main challenge here was twofold: a) providing a resilient service across the whole DARIAH infrastructure and b) selecting a sustainable solution with the available resources provided by the service provider, Huma-Num.

5.1 Huma-Num

Huma-Num is a Very Large Research Infrastructure (French: TGIR, Très Grande Infrastructure de Recherche) led by the French Ministry of Higher Education and Research and operated by the CNRS (the French National Centre for Scientific Research). It provides services to the entire Humanities and Social Sciences (HSS) academic community, particularly digital services focused on research data management that aim to help researchers manage the lifecycle of their data.

Through consortia of actors in scientific communities, Huma-Num supports the coordination of the collective production of corpora of sources with scientific recommendations and technological best practices. It also provides research teams with a range of utilities to facilitate the processing, access, storage, and interoperability of various types of digital data. This set of shared services comprises a grid of services, a platform for unified access to data (ISIDORE[29]), and long-term archival facilities. Huma-Num also

[29] Accessed July 9, 2020, http://rechercheisidore.fr/.

produces technical guides to good practice for researchers and, on occasion, conducts expertise and training initiatives.

Technically, the infrastructure itself is hosted in a large computing centre in Lyon[30]. A long-term preservation facility hosted at the CINES data centre[31] based in Montpellier is also used. In addition, a group of correspondents in the MSH[32] (Maison des Sciences de l'Homme) network all over France is in charge of relaying information about Huma-Num's services and tools.

5.2 Technical Requirements

Moving from a working prototype that demonstrates an idea to an engineered service is a complex process. We refer to engineering as implementing best practices to make the software ready to use without any previous knowledge. This implies tackling the following tasks:

- license checking (for open-source projects)
- consolidation of the project life cycle (build, testing, and deployment)
- publication of the documentation (for users, developers, and maintainers)
- definition and measurement of nonfunctional requirements, performances, scalability
- publication of metrics evaluating the ML-based system, using known corpora in order to provide measures comparable to the state of the art
- definition and measurement of functional requirements, such as expected behaviors and API responses via unit and integration tests

[30] Centre de Calcul de l'IN2P3 (CC-IN2P3), accessed July 9, 2020, https://cc.in2p3.fr/.

[31] C.I.N.E.S. (Centre Informatique National de l'Enseignement Supérieur), accessed July 9, 2020, https://www.cines.fr/.

[32] Réseau National MSH, accessed July 9, 2020, http://www.msh-reseau.fr/.

During the HIRMEOS project, the application was published on GitHub and released with an open source license (Apache 2 license[33]), a process that included replacing or rewriting parts that used libraries with incompatible licenses.

Entity-fishing is designed for fast processing on text and PDF documents, with relatively limited memory, and to offer relatively close to state-of-the-art accuracy (as compared with other NERD systems). The accuracy f-score for disambiguation is currently between 76.5 and 89.1 on standard datasets (ACE2004, AIDA-CONLL-testb, AQUAINT, MSNBC) as presented by Patrice Lopez at WikiDataCon in October 2017 (Lopez 2017) (table 1).

Table 1. Accuracy measures

	Priors	entity-fishing	Wikifier	DoSeR	AIDA	Spotlight	Babelfy	WAT	(Ganea & Hofmann, 2017)
ACE 2004	83.1	83.5	83.4	**90.7**	81.5	71.3	56.1	80.0	88.5
AIDA CONLL-testb	66.1	76.5	77.7	78.4	77.4	59.3	59.2	84.3	**92.2**
AQUAINT	80.3	**89.1**	86.2	84.2	53.2	71.3	65.2	76.8	88.5
MSNBC	71.1	86.7	85.1	91.1	78.2	51.1	60.7	77.7	**93.7**

The objective, however, is to provide a generic service that has a steady throughput of 500–1,000 words per second or one PDF page of a scientific article in 1–2 seconds on a mid-range (4 cores, 3 GB RAM) Linux server.

In section 4.3 we analyzed the application extensively, identifying the more critical parts. The engine contains two machine learning models (for ranking and selection) that are implemented through Smile[34] (Statistical Machine Intelligence and Learning Engine), a library written in Java/Scala providing an implementation of each Machine Learning algorithm (Random Forest, Classification, etc.). The storage is handled using a key-value database with memory-mapped files, Lightning Memory-Mapped Database (LMDB), which is available as a Java library called lmdbjndi (see sec. 4.3.2.2).

[33] Apache license version 2.0, January 2004, accessed July 9, 2020, https://www.apache.org/licenses/LICENSE-2.0.

[34] Accessed July 9, 2020, https://haifengl.github.io.

5.3 Deployment

The Huma-Num infrastructure's services and utilities[35] cover various sets of needs: storage, dissemination, processing, and archiving. As part of the hosting and dissemination services, they provide two types of solutions: the shared web cluster and virtual machine environments.[36]

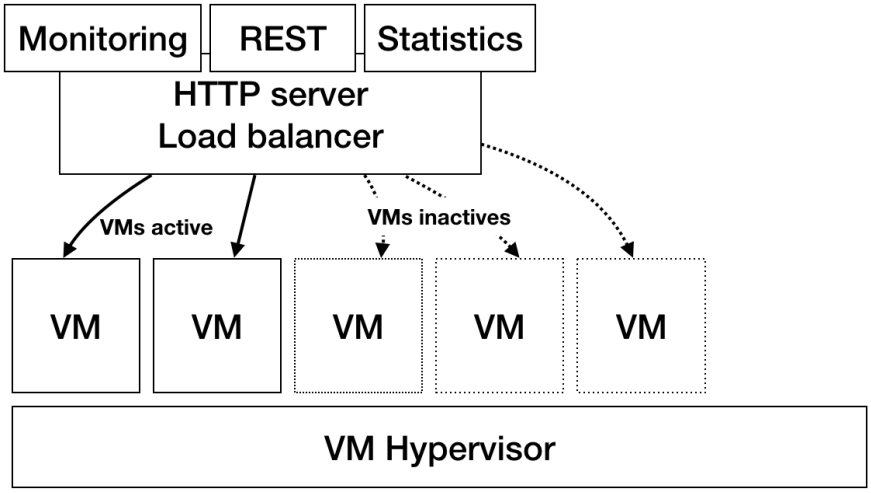

Figure 7. Virtual machine configuration

The web cluster is a shared solution hosting pre-configured CMSs (Omeka, Wordpress, Drupal, etc.), websites, and java web applications (running with tomcat, Jetty, BaseX, etc.). Users requesting a service obtain a ready-to-use application. The virtual machines (VMs), on the other hand, are intended for a more technical audience seeking greater flexibility or

[35] "Services et outils," Huma-Num, accessed July 9, 2020, https://www.huma-num.fr/services-et-outils.

[36] See Joel Marchand, "Huma-NUM la TGIR des humanités numériques," presentation at les Assises du CSIESR 2017 (the CSIESR Conference 2017), accessed July 17, 2020, http://assises2017.csiesr.eu/programme-1.

needing to deploy applications with special requirements. VMs are preconfigured with Ubuntu, Debian, or even Windows (fig. 7).

In order to meet our needs we have installed *entity-fishing* in a VM. Huma-Num provides normal HDD (hard disk storage) in two flavors, a fast-computing NAS and a distributed storage system.

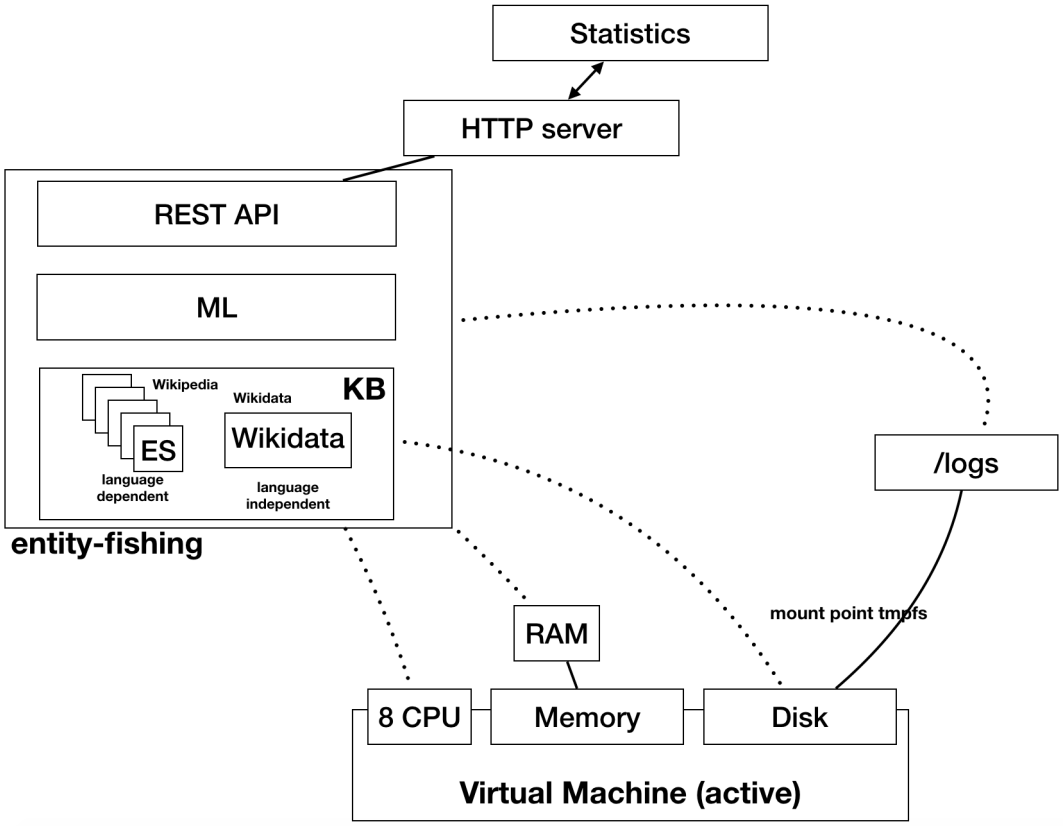

Figure 8. Final service configuration

We measured performance as *runtime* and *throughput* using JMeter [37], an open-source toolbox for simulating access load from users. We performed two types of tests, using PDF documents ranging from 10 to 200 pages and text paragraphs with an average size of 1,000 words. The tests ran for 20 minutes and 1 hour, using plain text content and PDF documents respectively. We repeated the tests using two different configurations: single-user and multi-

[37] Accessed July 10, 2020, https://jmeter.apache.org.

user (five users). We then recorded performances by measuring the server computation time and the throughput for each request, as illustrated in table 2.

Table 2. Runtime performances and throughput measured through sequential (single-user) or parallel requests (multiusers)

Scenario	Average runtime	Throughput
Text		
Single-user	0.675 s	1,265 char/s
Multiuser (5 users)	0.468 s	8,760 char/s
PDF		
Single-user	20.5467 s	1.1 pages/s
Multiuser (5 users)	21.5349 s	5.2 pages/s

The service was officially deployed in September 2017 after two months of tests. The initial configuration was set up with a virtual machine having 8 cores, 32 GB of RAM, and 100 GB of fast hard drive.

The service is accessible through an HTTP Apache 2 reverse proxy allowing long requests up to a maximum timeout of 30 minutes. Asynchronous requests would streamline the process (for more details, see sec. 6).

The infrastructure is monitored via an external service (on Uptime Robot[38]), which provides a monitoring dashboard[39] (fig. 9) and an email notification system for downtime.

[38] Accessed July 10, 2020, https://uptimerobot.com/.

[39] VM-huma-num status page for entity-fishing@huma-num, accessed July 10, 2020, https://stats.uptimerobot.com/nRyO1tpDV/779345024.

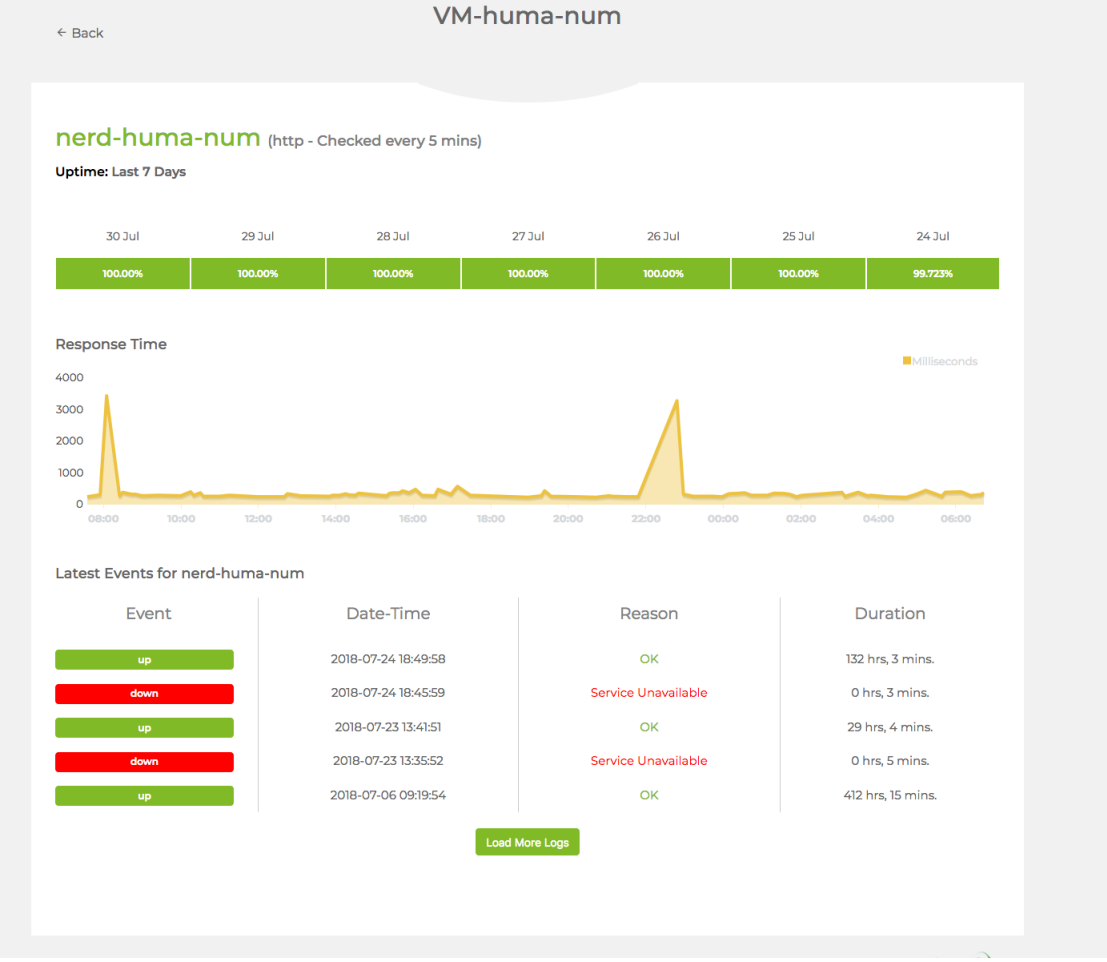

Figure 9. Monitoring page

We record usage statistics using AWStats[40] (fig. 10). By the end of 2018 more than 7 million documents were processed by *entity-fishing*.

[40] Accessed July 10, 2020, https://www.awstats.org/.

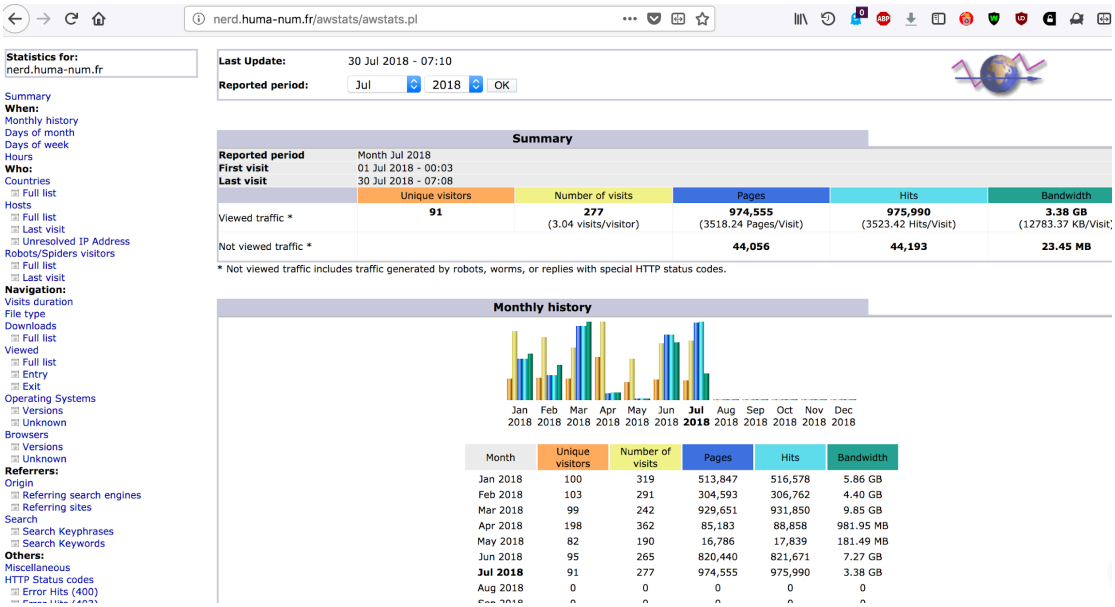

Figure 10 Statistics page

The HIRMEOS partners have chosen to integrate the application into their open access platform.

5.4 *Entity-fishing* Integrations

We have deployed *entity-fishing* for several use cases in collaboration with the HIRMEOS project partners OAPEN, OpenEdition, EKT, Göttingen University Press, and Ubiquity Press. Questions, considerations, and problems emerged when external partners started to use the resulting annotations according to their own needs and practices.

The two main implementations were the faceting and the entity visualization, integrated into the partners' already existing search interfaces as a facet search (fig. 11) or a word cloud over the whole collection (fig. 12).

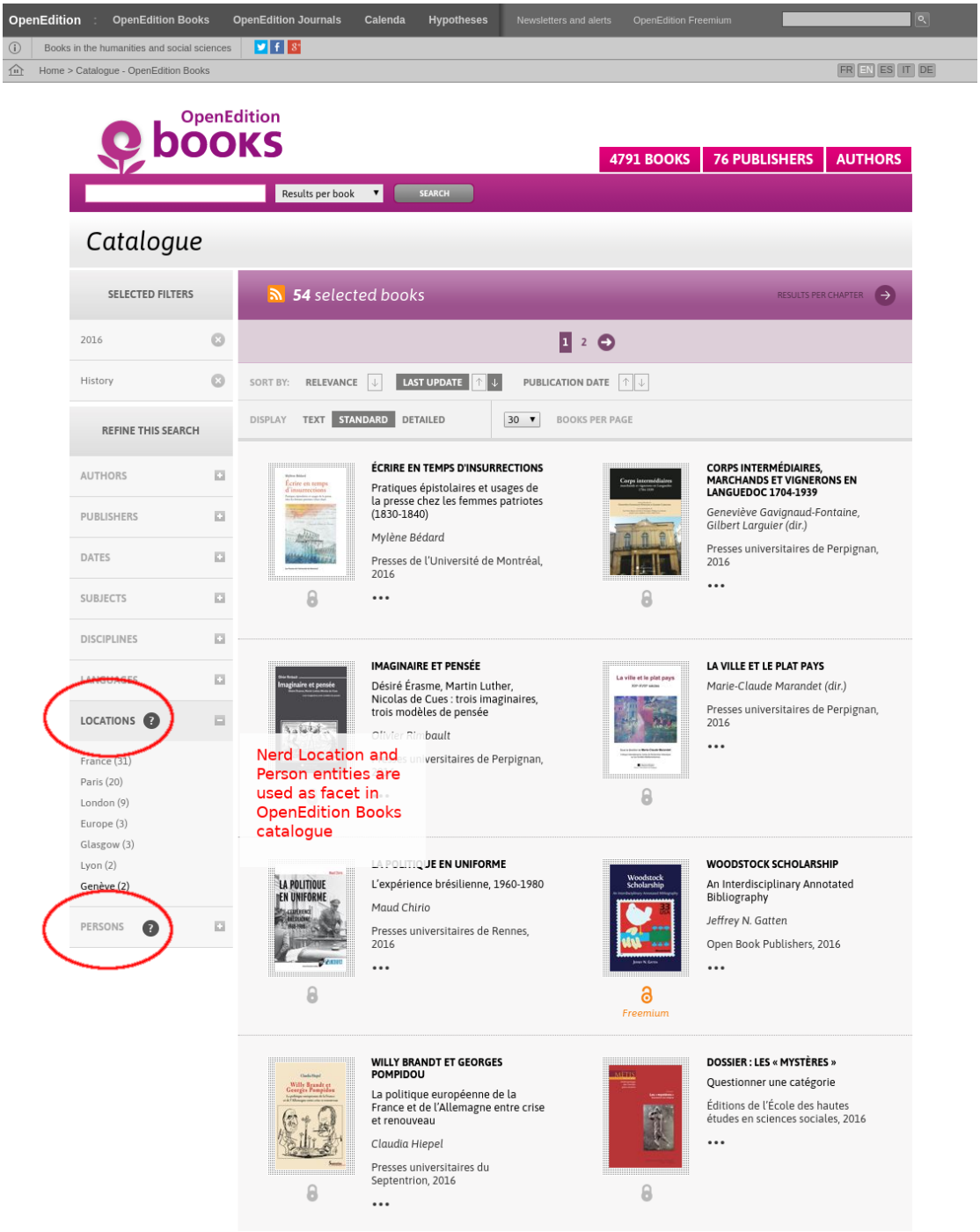

Figure 11. Use case implementation by Open Edition. Locations and Places automatically extracted from the content, using entity-fishing are displayed as search facets.

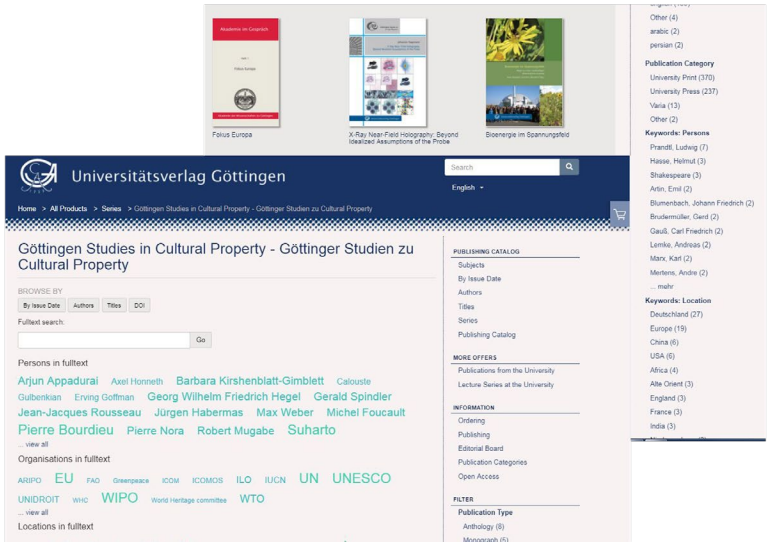

Figure 12. Use case implementation by Göttingen State Library

6 Conclusions and Future Work

The *entity-fishing* service has become an essential asset for the online delivery of the corpus of scholarly monographs. This will be particularly important when the OPERAS research infrastructure is set up to make more and more collections available to the research community. In terms of the technical deployment itself, we have provided all the necessary components of a sustainable service:

- release and publication of *entity-fishing* as open-source software[41]
- deployment of the service in the DARIAH infrastructure through Huma-Num[42]
- publication of evaluation data and metrics for content validation
- integration of the service with third-party platforms

[41] Accessed July 10, 2020, http://github.com/kermitt2/nerd..

[42] Accessed July 10, 2020, http://www.huma-num.fr/.

We still have work ahead of us to improve the accuracy of the disambiguation scores, which is currently floating just below the level of state-of-the-art performance. In fact, *entity-fishing* could easily be extended to support more languages.

Removing or deprecating the relatedness in favor of alternative, less computing-intensive techniques could reduce the impact of the storage on performance. On the subject of disambiguation, there has already been some interesting work on alternative Named Entity Recognition (NER) recognizer models using deep learning techniques[43]. Finally, the API could implement an asynchronous mechanism that handles large-scale computing more effectively.

7 Acknowledgments

We would like to address our warmest thanks to Patrice Lopez, who designed, demonstrated, and implemented the first version of *entity-fishing* during the CENDARI H2020 project. Patrice is also the author of GROBID[44] (GeneRation Of BIbliographic Data) (Lopez 2009), a machine-learning library for extracting, parsing, and restructuring raw documents such as PDF into structured TEI-encoded documents. We would like to thank our colleagues within the HIRMEOS project, especially Open Edition and the University of Göttingen State Library, for the particular support they provided in testing and disseminating *entity-fishing*. Finally, we would like to thank Huma-Num for hosting the service within their infrastructure.

References

Banski, Piotr, Bertrand Gaiffe, Patrice Lopez, Simon Meoni, Laurent Romary, Thomas Schmidt, Peter Stadler, and Andreas Witt. 2016. *Wake Up, StandOff!* Paper presented at the TEI Conference and Members' Meeting 2016, Vienna, Austria, September 26–30. https://hal.inria.fr/hal-01374102.

[43] DeLFT (Deep Learning Framework for Text) GitHub repository, accessed July 10, 2020, https://github.com/kermitt2/delft.

[44] GROBID GitHub repository, accessed July 17, 2020, https://github.com/kermitt2/grobid.

Brando, Carmen, Francesca Frontini, and Jean-Gabriel Ganascia. 2016. "REDEN: Named Entity Linking in Digital Literary Editions Using Linked Data Sets." *Complex Systems Informatics and Modeling Quarterly*, no. 7: 60–80. doi:10.7250/csimq.2016-7.04.

Buddenbohm, Stefan, and Raisa Barthauer. 2017. "D 4.1 - Gap Analysis of DARIAH Research Infrastructure." DARIAH research report.. https://hal.archives-ouvertes.fr/hal-01663594.

Cucerzan, Silviu. 2007. "Large-Scale Named Entity Disambiguation Based on Wikipedia Data." In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 708–16. Stroudsburg, PA: Association for Computational Linguistics. https://www.aclweb.org/anthology/volumes/D07-1/.

Edwards, Paul N. 2003. "Infrastructure and Modernity: Force, Time, and Social Organization in the History of Sociotechnical Systems." In *Modernity and Technology*, edited by Thomas J. Misa, Philip Brey, and Andrew Feenberg, 185–225. Cambridge, MA: MIT Press.

Lopez, Patrice. 2009. "GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction for Scholarship Publications." In *Research and Advanced Technology for Digital Libraries: 13th European Conference, ECDL 2009…: Proceedings*, edited by Maristella Agosti, José Borbinha, Sarantos Kapidakis, Christos Papatheodorou, and Giannis Tsakonas, 473–74. Lecture Notes in Computer Science 5714. Berlin, Heidelberg: Springer.

Lopez, Patrice. 2017. "Entity-Fishing." Slides presented at WikiDataCon 2017, Berlin, Germany, October 28–29. Last revised 8 February 2018, https://www.wikidata.org/wiki/Wikidata:WikidataCon_2017/Documentation; accessed July 11, 2020, https://grobid.s3.amazonaws.com/presentations/29-10-2017.pdf.

Lopez, Patrice, Alexander Meyer, and Laurent Romary. 2014. "CENDARI Virtual Research Environment & Named Entity Recognition Techniques." Poster presented at the conference Grenzen überschreiten – Digitale Geisteswissenschaft heute und morgen, Berlin, Germany, February 28, 2014. Einstein-Zirkel Digital Humanities. https://hal.inria.fr/hal-01577975.

Milne, David N., Ian H. Witten, and David M. Nichols. 2007. "Extracting Corpus Specific Knowledge Bases from Wikipedia." Working paper series, no. 03/2007, Department of Computer Science, University of Waikato, Hamilton, New Zealand. https://hdl.handle.net/10289/69.

Nadeau, David, and Satoshi Sekine. 2007. "A Survey of Named Entity Recognition and Classification." In *Named Entities: Recognition, Classification and Use*, edited by Satoshi Sekine and Elisabete Ranchhod [*Lingvisticae Investigationes* 30:1], 3–26. [Amsterdam and Philadelphia]: John Benjamins. doi:10.1075/li.30.1.03nad.

Pellissier Tanon, Thomas, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. 2016. "From Freebase to Wikidata: The Great Migration." In *WWW '16: Proceedings of the 25th International Conference on World Wide Web*, 1419–28. Geneva, Switzerland: International World Wide Web Conferences Steering Committee. doi:10.1145/2872427.2874809.

Ratinov, Lev, Dan Roth, Doug Downey, and Mike Anderson. 2011. "Local and Global Algorithms for Disambiguation to Wikipedia." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 1:1375–84. Stroudsburg, PA: Association for Computational Linguistics. https://www.aclweb.org/anthology/P11-1138/.

Romary, Laurent, and Jennifer Edmond. 2017. "Sustainability in DARIAH." Presentation at Sustainability of Digital Research Infrastructures for the Arts and Humanities (Workshop at the DARIAH Annual Event), Berlin, Germany, April 27. https://hal.inria.fr/hal-01516487.

Salton, Gerard, and Michael J. McGill. 1983. *Introduction to Modern Information Retrieval*. New York: McGraw-Hill.

Smith, David A., and Gregory Crane. 2001. "Disambiguating Geographic Names in a Historical Digital Library." In *Research and Advanced Technology for Digital Libraries: 5th European conference, ECDL 2001…: Proceedings*, edited by Panos Constantopoulos and Ingeborg T. Sølvberg, 127–36. Lecture Notes in Computer Science 2163. Berlin: Springer.

Steiner, Thomas. 2014. "Bots vs. Wikipedians, Anons vs. Logged-Ins (Redux): A Global Study of Edit Activity on Wikipedia and Wikidata." In *OpenSym '14: Proceedings of the International Symposium on Open Collaboration*, 25:1–25:7. New York: ACM. doi:10.1145/2641580.2641613.

TEI Consortium. 2020. TEI P5: Guidelines for Electronic Text Encoding and Interchange. Version 4.0.0. Last updated February 13, 2020. N.p.: TEI Consortium. https://tei-c.org/Vault/P5/4.0.0/doc/tei-p5-doc/en/html/.

Vanden Daelen, Veerle, Jennifer Edmond, Petra Links, Mike Priddy, Linda Reijnhoudt, Václav Tollar, and Annelies van Nispen. 2015. "Sustainable Digital Publishing of Archival Catalogues of Twentieth-Century History Archives." Paper presented at the conference Open History: Sustainable Digital Publishing of Archival Catalogues of Twentieth-Century History Archives, Brussels, Belgium, December 9–12. https://hal.inria.fr/hal-01281442.

Vrandečić, Denny. 2012. "Wikidata: A New Platform for Collaborative Data Collection." In *Proceedings of the 21st International Conference on World Wide Web*, 1063–64. WWW '12 Companion. New York: ACM. doi:10.1145/2187980.2188242.

Witten, Ian H., and David N. Milne. 2008. "An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links." In *Wikipedia and Artificial Intelligence: An Evolving Synergy. Papers from the AAAI Workshop*, pp. 25–30. Technical Report WS-08-15. [Palo Alto, CA]: AAAI Press. https://www.aaai.org/Papers/Workshops/2008/WS-08-15/WS08-15-005.pdf; author's version, https://hdl.handle.net/10289/1777.

Wright, Jan. 2012. "The Devil Is in the Details: Indexes versus Amazon's X-Ray." *The Indexer* 30 (1): 11–16. doi:10.3828/indexer.2012.4.