Semantic Web 1 (0) 1–5
IOS Press

Orbis: Explainable Benchmarking of Information Extraction Tasks

Adrian M.P. Braşoveanu [a,*], Albert Weichselbraun [b], Roger Waldwogel [b], Fabian Odoni [b], and Lyndon J.B. Nixon [a]

[a] *Modul Technology, Modul Technology GmbH, Am Kahlenberg 1, 1190 Vienna, Austria*
E-mails: adrian.brasoveanu@modul.ac.at, lyndon.nixon@modul.ac.at
[b] *Swiss Institute for Information Research, University of Applied Sciences of the Grisons, Switzerland*
E-mails: albert.weichselbraun@fhgr.ch, roger.waldwogel@fhgr.ch, fabian.odoni@fhgr.ch

Editors: First Editor, University or Company name, Country; Second Editor, University or Company name, Country
Solicited reviews: First Solicited Reviewer, University or Company name, Country; Second Solicited Reviewer, University or Company name, Country
Open reviews: First Open Reviewer, University or Company name, Country; Second Open Reviewer, University or Company name, Country

Abstract. Competitive benchmarking of information extraction methods has considerably advanced the state of the art in this field. Nevertheless, methodological support for *explainable benchmarking*, which provides researchers with feedback on the strengths and weaknesses of their methods and guidance for their development efforts, is very limited. Although aggregated metrics such as F1 and accuracy support comparison of annotators, they do not help in *explaining* annotator performance.

This work addresses the need for explainability by presenting Orbis, a powerful and extensible explainable evaluation framework which supports drill-down analysis, multiple annotation tasks and resource versioning. It, therefore, actively aids developers in better *understanding* evaluation results and identifying shortcomings in their systems.

Orbis currently supports four information extraction tasks: content extraction, named entity recognition, named entity linking and slot filling. This article introduces a unified formal framework for evaluating these tasks, presents Orbis' architecture, and illustrates how it (i) creates simple, concise visualizations that enable visual benchmarking, (ii) supports different visual classification schemas for evaluation results, (iii) aids error analysis, and (iv) enhances interpretability, reproducibility and explainability of evaluations by adhering to the FAIR principles, and using lenses which make implicit factors impacting evaluation results such as tasks, entity classes, annotation rules and the target knowledge graph more explicit.

Keywords: Content Extraction, Named Entity Linking, Named Entity Recognition, Slot Filling, Knowledge Graph, Benchmarking, Corpus Quality

1. Introduction

Applying complex information extraction (IE) pipelines to the web is considered challenging since web pages exhibit a high variability in layout, structure, noise and information content and, therefore, require content extraction (CE), named entity recognition (NER), named entity linking (NEL) and slot filling (SF) components to work in concert, to obtain reasonable performance.

Consequently, benchmarking information extraction pipelines requires considering these subtasks and their impact on each other. While literature extensively covers some of these tasks (e.g., named entity recognition (NER) and named entity linking (NEL), others such as content extraction are rarely discussed.

Barbesi & Lejeune [3], for example, benchmarked the performance of content extraction tools, but we are not aware of any system that support researchers in evaluating context extraction tasks from unstructured web pages or forums.

*Corresponding author. E-mail: adrian.brasoveanu@modul.ac.at.

NER focuses on the identification of named entities, whereas NEL links entities to a knowledge graph (KG) like DBpedia or Wikidata. Identifying entities and relations is a well-studied problem and supported by several benchmarking tools. Past competitions have been instrumental in advancing research in information extraction methods, in particular for NEL where competitions complemented with advanced evaluation toolkits such as neleval [13] and GERBIL [30] have triggered significant improvements of NEL systems. Latest reports indicate accuracy values of up to 95% on some datasets [2], although the differences between results on various datasets can be as high as 40%.

However, these results disguise the differences that occur in NEL due to multiple factors such as the different entity (sub)classes, the annotation rules, and the target knowledge graph. Current systems are fine-tuned for accuracy in a limited set of NEL tasks which are often only considered isolated.

Tasks such as SF which mines additional information on entities, and Event Argument Linking (EAL) which is concerned with the detection of events and arguments, currently do not have the same level of tool support, although some competitions like TAC-KGP [19] include these tasks, whereas others (e.g., OKE [25] or SemEval tasks [23]) focus on a subset of them. In addition, most evaluation systems only consider isolated tasks rather than whole information extraction pipelines.

Another significant shortcoming of current benchmarking systems is their lack of support for drill-down analyses that help in explaining error types, either visually or through text cues. Such analyses have the potential to considerably improve the efficiency of research and development in information extraction methods, since they help in uncovering flaws which might have been caused by factors such as dataset errors, KG errors, automated annotation errors, and even evaluation scorer errors [5].

A toolkit that supports holistic benchmarking of information extraction tasks (i.e., considering the whole pipeline), debugging and visualizing errors and in explaining them, would provide significant advantages towards improving the performance of annotators for complex information extraction tasks. The research presented in this paper has been motivated by this need to create and optimize sophisticated information extraction pipelines for applied research projects such as InVID [1], ReTV [2], DISCOVER [3], Job-Cockpit [4], and CareerCoach [5] which had been seriously impaired by:

1. the lack of benchmarking tools covering the whole information extraction process, and
2. missing support for drill-down analysis which significantly prolonged development cycles.

Consequently, we have developed an extensible benchmarking ecosystem called Orbis[6], which allows researchers to extend the coverage of the information extraction process by creating plugins, and provides analytics and visualizations specifically designed towards promoting interpretability and explainability. The first version of Orbis was created to support the development of Recognyze, a graph-based named entities disambiguation tool [38]. Once Recognyze entered production, the tool was expanded to support more IE tasks. Orbis supports standard evaluation metrics (e.g., precision, recall, F1, accuracy), drill-down analyses by visualizing gold standard and system results in context (i.e., the annotated document), modes for comparing multiple evaluations (i.e., the gold standard and the output of two or more information extraction systems), and overview pages which highlighting significant changes between evaluation runs. These tools aid experts in quickly identifying shortcomings within their methods and in addressing them. Comparative evaluations can be used to outline differences between systems, evaluation settings, and gold standard versions.

The remainder of this paper is organized as follows: Section 2 discusses the state of the art; Section 3 introduces the theoretical framework used for performing evaluations of content extraction (CE), named entity recognition (NER), named entity linking (NEL), and slot filling (SF); Section 4 provides a brief overview of Orbis, its main components and capabilities; and Section 5 discusses explainable benchmarking principles and illustrates how Orbis implements them. The paper concludes with a discussion in Section 6 and an outlook and conclusions presented in Section 7.

[1] https://www.invid-project.eu/
[2] https://retv-project.eu/
[3] https://fhgr.ch/discover
[4] https://fhgr.ch/job-cockpit
[5] https://fhgr.ch/CareerCoach
[6] Orbis and related repositories are available under the following GitHub namespace: https://github.com/Orbis-eval/

2. Related Work

Ling et al. [22] describe early issues found in NEL evaluations, such as the lack of (i) clear task definitions; (ii) evaluation tools that use the same datasets; and (iii) means for understanding which aspects make certain systems better than others. Benchmarking systems such as GERBIL [30] help in addressing these concerns, although their suitability for providing clear task definitions and support in understanding the strengths and weaknesses of systems is still hotly debated, as illustrated by [17, 32].

2.1. Black-box Evaluation

The tools included in this category typically provide only evaluation results and no additional entity-level explanations.

Cornolti's BAT framework [9], an automated evaluation system that measures per-task performance, defined new evaluation types based on the content of the annotation tasks. Six annotation tasks have been initially included in the BAT framework: Disambiguate to Wikipedia (D2W), Annotate to Wikipedia (A2W), Scored-annotate to Wikipedia (Sa2W), Concepts to Wikipedia (C2W), Scored concepts to Wikipedia (Sc2W) and Ranked concepts to Wikipedia (Rc2W).

GERBIL [30] and its extensions [35, 36] were designed to support multiple experiment types using black box evaluation techniques. GERBIL was initially designed to improve upon the BAT framework. It has become one of the most popular evaluation tools since it provides: (i) many datasets and annotators for experiments; (ii) a sizeable number of experiment types; (iii) a public version that allows access to all performed experiments via a unique identification number; (iv) the development of public competitions around it (e.g., OKE Challenges). Criticism against GERBIL has initially revolved around: (i) its black box evaluation approach which has not provided system developers with sufficient insights to improve their own tools; (ii) lack of a query interface to discover past experiment results for the general public; and (iii) extensibility.

Some recent versions have mainly been targeted at improving GERBIL's extensibility. An early attempt at performing typed evaluations [35] has led to improved filtering mechanisms for the various experiment types (e.g., filters for entity types, pagerank and hitsscores). The follow-up work [36] dedicates ample space to understanding a variety of phenomena like missing annotations, popularity or the likelihood of confusion of entities and surface forms using GERBIL. While such improvements clearly address crticism of GERBIL's extensibility, the other two points remain valid at the time of writing.

Facebook's KILT [28], a toolkit for evaluating shared tasks that rely on a common knowledge graph for grounding, provides both general and task-specific evaluations, but does not offer comprehensive explanations for them. Besides some basic interface for navigating the tasks and providing the results, the tool does not provide detailed explanations.

Some basic tools for the evaluation of web pages cleaning also exist, cleaneval [4] and Waddle[7] [3] being the most well-known, but unfortunately they are not tailored to specific tasks like forum extraction and all of them are black-box tools.

2.2. Explainable Evaluation

The neleval[8] system [13] based on the TAC-KGP guidelines provides primary error explanations. As an alternative, visual evaluation systems such as VEX [15] also allow close inspection of the evaluation results and support designers in improving system performance.

Tools that aid in identifying errors in gold standards can also be included in this category, since they help explain the various results obtained on old gold standards. EAGLET [18], a tool designed to help improve reference standards, was built around a pipeline that draws upon the results returned by the annotators supported by GERBIL and includes an error detection module focused on several error types (e.g., wrong position, overlaps, combined entities, etc). The precursor of Orbis used a collection of scripts built on top of the neleval suite [5] that used four annotators and focused mostly on identifying major error classes (e.g., KG, DS errors, etc). The NIFify tool [31] was developed to help create NIF datasets but can also be used to validate annotated datasets and help in spotting various errors. The main problem with most of these tools is that there is no unified error taxonomy, and therefore errors classified in a particular category by one of the tools might be classified under a different category or not classified by another tool.

While most of the tools for evaluating IE tasks have not been designed with explainable evaluation in mind,

[7] https://github.com/rundimeco/waddle
[8] https://github.com/wikilinks/neleval

examples of such tools can be found in other areas of Computer Science. A good description of a toolkit used for visualizing challenge results for image analysis competitions is presented in [40]. The toolkit contains a wide array of visualizations including, but not limited to, box plots, ranking heatmaps, line plots, blob plots or significance maps. Each graph is used to showcase a different aspect of the system performance, as ranking heatmaps, for example, can easily show which system performs best, whereas blob plots would explain which system is most stable through the various tasks.

3. Background and Problem Statement

As described in Section 2, explainability is key towards improving information extraction methods. Before describing methods for adding explainability layers into benchmarking systems, this section provides background on formal problem statements for the following IE tasks:

1. *content extraction* ensures that only relevant content is extracted for analysis,
2. *named entity recognition* locates mentions of named entities in textual content and determines the entity type(s),
3. *named entity linking* identifies links to mentioned named entities within target knowledge graphs, and
4. *slot filling* extracts information on relations in which the named entities participate.

3.1. Content extraction

Apart from web data accessible via APIs or structured formats such as RSS and JSON-LD, web pages, forums and social media typically provide a mixture of unstructured content such as the relevant text, associated images or videos as well as noisy elements (e.g., links to other content, irrelevant text, advertisements, etc.) which have the potential to seriously impact the results of downstream IE tasks.

3.1.1. Problem Statement

Content extraction addresses this issue by identifying and extracting relevant content from web sources in a form suitable for the subsequent processing steps. *Boilerplate removal*, for instance, is a content extraction method that removes boilerplate elements from media articles and blog posts. *Forum extraction*, in contrast, is concerned with identifying posts in web forums where a typical web page contains multiple posts structured in different ways (topics, threads, replies, etc.). Optionally, forum extraction may also include the post metadata such as author, date and post structure [37]. Forum extraction methods can also be applied for social media, especially if the considered pages have various conversations with a thread structure.

Definition. *A content extraction system analyzes an input document d_i that contains information which is encoded as plain text, markup (e.g., XML-based formats, markdown etc.) or in a binary format (e.g., JPEG, MP4, PDF, DOCX etc.) to extract one or more text strings S_i^r relevant to downstream IE processes.*

3.1.2. Evaluation metrics

The performance of context extraction systems is computed by comparing the extracted text strings S_i^r with strings obtained from gold standard annotations S_i^g. Metrics may either perform string-based (e.g., Levenshtein distance) or token-based (e.g. Jaccard Coefficient and token-based similarity) comparisons. Token-based methods split text strings into tokens t_j to compare the extracted token set T_i^r to the tokens T_i^g within the gold standard text. Currently, Orbis supports the following three evaluation metrics:

1. The *normalized Levenshtein distance* (lev_n^i) is a string-based metric that normalizes the Levenshtein distance with the length of the extracted text ($|S_i^r|$) and the corresponding gold standard annotation ($|S_i^g|$) as outlined below:

$$lev_n^i(S_i^r, S_i^g) = \begin{cases} 0 \text{ if} |S_i^r| = 0 \text{ or } |S_i^g| = 0 \\ \frac{lev(S_i^r, S_i^g)}{max\left(|S_i^r|, |S_i^g|\right)} \text{ otherwise} \end{cases} \quad (1)$$

   Text snippets that have been missed ($|S_i^r|$=0) and superfluous text ($|S_i^g|$=0) yield the minimum similarity of 0 since $|T_i^r \cap T_i^g| = 0$ in these cases.
2. The *Jaccard Coefficient* (J) is a token-based metric which is computed as outlined below:

$$J(T_i^r, T_i^g) = \frac{T_i^r \cap T_i^g}{T_i^r \cup T_i^g} \quad (2)$$

3. The *token-based similarity* uses the token overlap to compute precision, recall and the F1 measure:

$$P(T_i^r, T_i^g) = \frac{|T_i^r \cap T_i^g|}{|T_i^e|} \quad (3)$$

$$R(T_i^r, T_i^g) = \frac{|T_i^r \cap T_i^g|}{|T_i^g|} \quad (4)$$

$$F_1(T_i^r, T_i^g) = \frac{P(T_i^r, T_i^g) \cdot R(T_i^r, T_i^g)}{P(T_i^r, T_i^g) + R(T_i^r, T_i^g)} \quad (5)$$

3.2. Named Entity Recognition (NER)

Named entity recognition is one of the oldest NLP tasks. The task was proposed in its current form in 1995 at the MUC-6 challenge [34]. The idea was to add markup that identifies entities and coreferences in text, as well as possible relations.

3.2.1. Problem Statement

Definition. *A Named Entity Recognition (NER) system takes an input string S_i^r, and parses it to identify and classify mentions of entities $m_{[s_i]}^{t_i}$ or $m_{[x_i,y_i]}^{t_i}$ with surface form s_i within the text into entity types $t_i \in T$. The variables x_i and y_i indicate the start and end position of the surface form s_i within the parsed text string.*

Although standard NER tasks only use a limited set of non-overlapping classes (e.g., LOC, ORG, PER and MISC), recent evaluations such as the *TAC-KBP 2019 Ultra-Fine-Grained Name Tagging for Entity Types* task considers 187 fine-grained types, and the task's designers have even set up an optional task with 7,309 YAGO/WordNet types [20]. In addition, fine-grained named entity recognition may also yield multiple types (e.g., `foaf:Person`, `yago:Politician-110450303`, `yago:President110467179`) for individual entities.

3.3. Named Entity Linking (NEL)

NEL is considered an extension of the NER task in the context of knowledge graphs that was proposed a decade after the MUC-6 conference by Bunescu and Pasca [6]. Regardless of the model that is globally used for disambiguation, NEL tools are required to link mentions of named entities to a target knowledge graph.

3.3.1. Problem Statement

Definition. *A Named Entity Linking system (also called automated annotator) links a mention $m_{[s_i]}^{e_i,KG}$ or $m_{[x_i,y_i]}^{e_i,KG}$ of a named entity with surface form s_i within a document d to the corresponding entity e_i in a knowledge graph (KG). The variable x_i indicates the mention's start position within the document and y_i the corresponding end position.*

Mentions may overlap and the specification of the knowledge graph can be omitted, if it is not relevant for the application (e.g., if we do not consider different KG versions in the given use case).

NEL systems distinguish between the following types of mentions:

1. $m_{[s_i]}^{e_i,KG}$ surface forms s_i that were linked to an entity e_i within a knowledge graph KG;
2. $m_{[s_i]}^{nil}$ mentions of Named Entities (NEs) that are not found in the KG and, therefore, are not linked (i.e., NIL entities);
3. $m_{[s_i]}^{\emptyset}$ candidate mentions with surface form s_i that do not refer to a named entity.

3.3.2. NEL Benchmarking Issues

Evaluating NEL performance is a complex task which involves multiple components:

- one or multiple NEL tools;
- a specific KG version (e.g., DBpedia 2016-10, Wikidata 25-04-2019, etc);
- one or multiple gold standards which are generally collections of documents annotated by human or automated annotators;
- a scoring system that will evaluate the automated annotators against the gold standards, classify the answers and score them.

The following sections outline how design decisions on these factors influence NEL benchmarking results and, therefore, impact the explainability and reproducibility of NEL evaluations.

3.3.3. Different Annotator Configurations

Optimizing annotators for a specific dataset is a common practice which would be unproblematic if optimizations were fully documented, enabling third-party researchers to reproduce annotator configurations. In practice, these configurations are often not explained which affects evaluations by yielding a different annotator behavior and consequently a different set of linked mentions:

$$\mathcal{M}^* = \left\{ m_{[s_1]}^{*,e_1,KG}, ..., m_{[s_n]}^{*,e_n,KG} \right\} \quad (6)$$

rather than the one obtained in the original evaluation:

$$\mathcal{M} = \left\{ m_{[s_1]}^{e_1,KG}, ..., m_{[s_l]}^{e_l,KG} \right\}. \quad (7)$$

3.3.4. Different Knowledge Graph Versions

Knowledge Graphs evolution poses another challenge to NEL benchmarking. Different Knowledge Graph versions:

– often differ in the available entities, and may (i) transform a former NIL entity $m^{nil}_{[s_i]}$ into a $m^{e_i,KG}_{[s_i]}$ link to the KG or vice versa, and (ii) add or remove ambiguities into the NEL linking process, therefore, linking an entity $m^{e_i,KG}_{[s_i]}$ to another KG entry e'_i.
– refine the available context information on named entities which in turn might influence the linking of named entities as described above.

It is important to note that KGs can go through consolidation phases, especially after periods of rapid growth. In such periods, links may be added or removed to consolidate the entire graph. Update frequency (or alternatively, the RDF dumps publishing frequency) is therefore an important property of a KG. Due to this, it is important that each dataset and annotator tool is tagged with the corresponding KG version.

Consequently, gold standard datasets as well as NEL tools should indicate the version of the target knowledge graph. Strategies for dealing with knowledge graph evolution are discussed in Section 5.3.2.

3.3.5. Different Gold Standard Annotation Guidelines

The huge variation in naming conventions allows for different options for annotating entities. Person names, for example, frequently include titles (*President*, *Senator*, *Prince*, *Judge*, *CEO*, etc.). Locations can include place qualifiers (e.g., N/E/S/W); abbreviations (e.g., *CA* for California); nested entities (e.g., *New York Stadium*); addresses (e.g., *221B Baker Street*). Fictional works frequently contain references to author names (e.g., *Dante's Inferno* and *Zeppelin's Immigrant Song*) or to the franchise to which they belong (e.g., *Star Trek Discovery*). Events are even more complex, since they sometimes take the name of the place or time when or where they happened (e.g., *Grenfell Tower* or *September 11*) and, therefore, they require more fine-grained disambiguation mechanisms and additional context.

Annotation guidelines need to be included when a gold standard is created to clarify how experts annotated the entities. In practice this leads to different annotation styles which in turn complicates the task of comparing results. The surface form "Star Trek Discovery"[9], for instance, can yield as named entities (depending on the used annotation style) either one of:

$$m^{dbr:Star\_Trek\_Discovery}_{[\text{Star Trek Discovery}]} \tag{8}$$

or

$$m^{dbr:Star\_Trek\_Discovery}_{[\text{Star Trek Discovery}]}, m^{dbr:Star\_Trek}_{[\text{Star Trek}]} \tag{9}$$

It can be seen that divergent annotation styles yield different gold standard annotations $\mathcal{M}$ and consequently also varying results from evaluations, again impacting the reproducibility.

3.3.6. Different Scoring Rules

Within the context of an evaluation, *scoring rules* outline the conditions under which a gold standard mention $m^c := m^{e_c,KG}_{[x_c,y_c]}$ and a mention returned by the NEL system $m^s := m^{e_s,KG}_{[x_s,y_s]}$ are considered equivalent to each other.

The following scoring rules are frequently used in NEL evaluations:

1. *perfect match* $\mathcal{P}$ - the entities refer to the same KG entity e_i, and exactly same surface form s_i.
2. *contained match* $\mathcal{C}$ - both entities refer to the same KG entity e_i and the surface form of the mention returned by the NEL system m^s is contained in the surface form of the corpus mention m^c, i.e., $x^s_i \geqslant x^c_i$ and $y^s_i \leqslant y^c_i$.
3. *overlapping match* $\mathcal{O}$ - this case is equivalent to the contained match but further relaxes the restrictions on the surface form, so that even an overlap (i.e., $y^s_i \geqslant x^c_i$ and $x^s_i \leqslant y^c_i$) between entities is considered a valid match.

The used scoring rule has a significant impact on the computation of the NEL system's performance metrics such as precision and recall.

Simply by enumerating these issues, it can easily be seen that explainability, reproducibility and quality of the current slate of evaluations still need a lot of work.

[9] The surface form corresponds to the DBpedia resource https://dbpedia.org/page/Star_Trek:_Discovery which will be represented through its abbreviated form dbr:Star_Trek_Discovery in this article.

3.4. Slot Filling

Slot filling can be considered a special case of relation extraction which requires the extraction of a predefined set of relations r_j (i.e., the slots) from textual content.

3.4.1. Problem statement

Definition *Slot filling collects information of the form (e_i, r_j, e_k) on entities e_i from text strings S_i^r that have been extracted from unstructured input documents d_i and can be used to create, extend or refine a knowledge graph KG [1].*

In this definition e_i refers to the entity on which information is collected, r_j denotes the slot on which information is collected (e.g., `per:employee_of`, `foaf:age`, `per:parents`), and e_k to the slot's value. Slot filling may, therefore, also be understood as a query that searches for entities e_k which are in relation r_j to entity e_i.

Slots may differ in cardinality and in the type of values they accept. The slot `foaf:age`, for instance, has a cardinality of one and only accepts a numeric value, `per:parents` has a cardinality of two and accepts entities of type Person, and `per:employee_of` usually has a cardinality $\geqslant 0$ and allows for entities of type Person and Organization.

Slot filling evaluations usually also consider multi-hop queries which require combining queries. For instance, filling the slot `per:birthday_of_children` for the entity *Albert Einstein* will require combining two slot filling tasks:

1. a query for the slot `per:has_child` which yields the three entities *Lieserl Einstein*, *Hans Albert Einstein* and *Eduard Einstein*, and
2. a query for slot `foaf:birthday` for these three entities which provides the answer for the given multi-hop query (*27 January*, *14 May* and *28 July*).

For the evaluation, the slot's values are still considered to be the results of one "virtual query" and are solely scored based on the correctness of the outcome [8]. False positives (e.g., birthdays that are not part of the correct answer) and negatives (e.g., birthdays that have been missed), are, therefore, scored as incorrect.

3.4.2. Evaluation metrics

Slot filling usually draws upon the standard evaluation metrics precision, recall, and F1 measure that are obtained by comparing the extracted slots (e_i, r_j, e_k) to the corresponding gold standard entries.

For a result to be considered a true positive (TP), the whole triple of (e_i, r_j, e_k) must match a gold standard entry. Superfluous results that do not match gold standard entries are considered false positives (FP), and missing triples count as false negatives (FN).

Finally, most evaluations only score triples that are related to the query, i.e., where r_j is the queried slot, e_i the entity used in the query and e_j the entity or value that occurs in the result set [8]. This restriction is important for slots that require multi-hop queries, since these queries may return additional triples that are not directly relevant to the answer.

Based on this definition, precision (P), recall (R) and the F1 measure are obtained as follows:

$$P = \frac{|TP|}{|TP \cap FP|} \tag{10}$$

$$R = \frac{|TP|}{|TP \cap FN|} \tag{11}$$

$$F1 = \frac{P \cdot R}{P + R} \tag{12}$$

Computing precision, recall and F1 measure based on the number of total TP, FP and FN across all evaluations yields micro precision (mP), micro recall (mR) and micro F1 (mF1) which refer to the average system performance to expect for an arbitrary query. In contrast, computing P, R and F1 measure for each slot, and then averaging the metric yields the macro metrics (MP, MR and MF1) which indicate the average per slot performance.

While these metrics provide a good summary of the system performance, they are not by themselves helpful for understanding and improving slot filling components. This is particularly true since slot filling requires coupling multiple components and is, therefore, heavily affected by pipeline effects where errors multiply across modules. Consequently, Orbis provides metrics and visualizations for evaluating upstream modules such as content extraction and named entity linking in addition to components for visualizing slot filling results.

4. Orbis Architecture

Orbis focuses on explainable benchmarking, i.e., on providing benchmarks, analytics and visualizations that help system designers in better understanding the strengths and weaknesses of their systems as well as explaining the reasons for a particular evaluation re-

sult. Its visual debugging capabilities, for instance, allow comparing results with both gold standard annotations, other tools and iterations of a particular NEL system. Orbis can also be used as a toolkit for building error classification schemes or new evaluation types.

4.1. Orbis Evaluation Pipeline

Orbis' architecture was developed around the idea of flexibility. Almost all important components are implemented as plugins and can be combined into use-case-specific evaluation pipelines. The plugins focused on visualizing corpora and results are particularly useful since they allow us to debug evaluations. The core Orbis interface, for instance, is document-centric and displays both the gold standard and the annotator result. Two viewing modes are available: *standard* and *dark* mode.

Evaluation pipelines are described by YAML configuration files and typically consist of the following three stages: acquisition, assessment and presentation.

1. The acquisition stage loads the data needed for the evaluation run, such as the gold standard and the predictions (i.e., the annotations provided by the NEL tool). The data object that holds all the data for an evaluation run is passed from stage to stage.
2. The assessment stage compares the gold standard to the predictions and produces a scored result, e.g., a confusion matrix, and the corresponding metrics like precision, recall and F1 score.
3. The presentation stage converts the evaluation results into different output formats and produces artifacts such as analytics, tables and visualizations to analyze and process the results.

Orbis also supports integrating external extension packages into the evaluation pipeline in each stage. External packages are not part of the core Orbis distribution, but installed separately using pip or other Python installation methods. Orbis will detect installed packages automatically and integrate these extensions into the corresponding stages of the pipeline.

Some useful extensions include:

- *repoman* for integrating new gold standards into the Orbis evaluation pipeline;
- annotator pipelines for ingesting data from automated annotators;
- *single_view* for examining corpora;
- *tunnelblick* displays corpus items in an HTML view;
- *satyanweshi* compares the predictions of two evaluation runs side by side;
- *legion* computes the agreements between multiple annotators for error classification tasks.

Table 1
Datasets integrated into Orbis.

Task	Dataset	Language	Description
CE	WebForum52	EN/DE	Medical forums
NER/NEL	N3 - Reuters128	EN	Short News
NER/NEL	N3 - RSS500	EN	Blogs
NER/NEL	N3 - News100	DE	News
NER/NEL	OKE 2016	EN	Abstract
NER/NEL	KORE50	EN	Short sentences
NER/NEL	DBpedia Spotlight	EN	News
NER/NEL	MediaRes100	EN	Wikinews
SF	JobCockpit	DE	HR extraction

Table 2
Annotators integrated into Orbis.

Task	Annotator	Languages	Formats
CE	Inscriptis [37]	Agnostic	JSON
CE	boilerpy [21]	Agnostic	JSON
CE	Dragnet [27]	Agnostic	JSON
CE	jusText [29]	Agnostic	JSON
CE	Harvest [37]	Agnostic	JSON
NER/NEL	DBpedia Spotlight [10]	EN,DE	NIF
NER/NEL	Babelfy [24]	EN,DE	NIF
NER/NEL	AIDA [16]	EN,DE	NIF
NER/NEL	FREME [12]	EN,DE	NIF
NER/NEL	OpenTapioca [11]	EN,DE	NIF
NER/NEL	Spacy [7]	EN,DE	NIF
NER/NEL	Recognyze [38]	EN,DE	NIF
SF	Recognyze [38]	EN,DE	NIF, JSON

4.2. Orbis Datasets and Annotators

In addition to custom datasets and annotators, Orbis provides built-in support for a number of popular gold standards and annotators which are listed in Table 1 and 2. NER/NEL/SF annotators were integrated using the NIF format, where possible, as it is considered the industry standard. For annotators that do not offer NIF output such as Spacy dedicated wrapper have been created. The CE annotators were integrated using the JSON format.

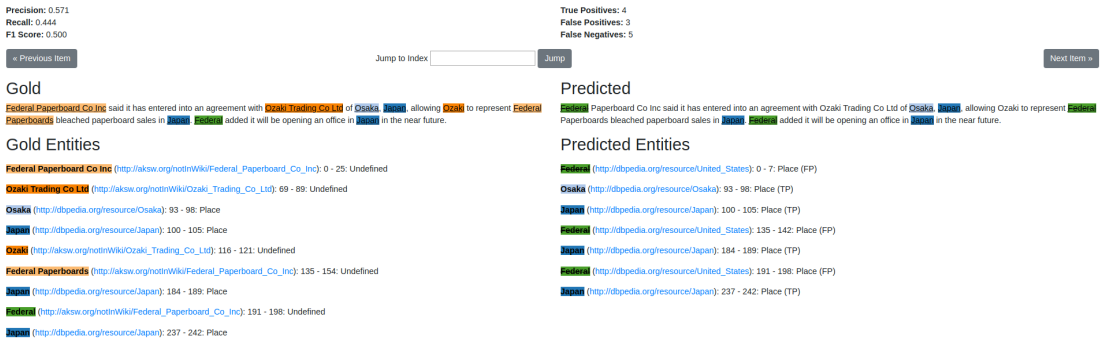

Fig. 1. A cropped screenshot of the results as generated by Orbis for item 21 of the Reuters128 evaluation corpus as annotated with Recognyze. Matching colors indicate identical resources.

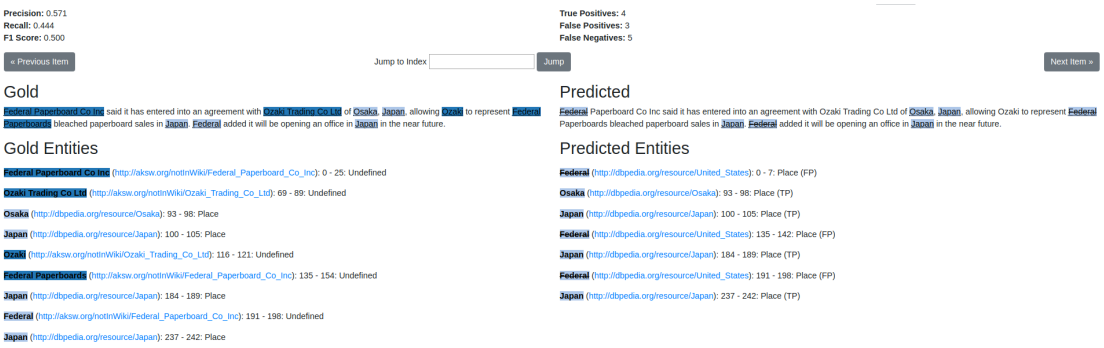

Fig. 2. A cropped screenshot of the types classification scheme for the Reuters128 corpus. Task: NEL

4.3. Orbis Visual Interface

Orbis visually displays gold annotations and annotator results in tabs next to each other. Color coding supports experts in quickly comparing named entities annotated in the gold standard to the annotator results. These visualizations aid in the rapid identification of errors in the annotator's results. Orbis also allows displaying multiple annotator results alongside the gold standard annotation to facilitate drill-down analysis that contrast different evaluation settings and systems. In addition, Orbis supports exporting results in multiple formats such as JSON or HTML.

The interface was designed based on the grammar of graphic principles outlined by Wilkinson [41], as well as on the two very influential taxonomies of Shneiderman [33] and Heer [14]. In general we follow the guidelines of visual analytics expressed by Shneiderman: *overview* first, *zoom*, *filter*, then provide the *details-on-demand*, *relations (relate principle)*, *history* and *extract* [33]. The *overviews* contain general corpus results (e.g., P, R, F1). Experts can then *zoom* into single documents or obtain *details-on-demand* on individual entities. The *relations (relate principle)* between various evaluation runs can easily be explored via the interface which allows comparing runs from the same or multiple annotators. This relational analysis typically constitutes the start of the fine-grained analysis phase. The *extract* phase is implemented in various parts of the interface via highlighting (e.g., in the overviews view), where all the problematic documents are visually marked.

Drill-down analyses on the level of individual annotations help in performing thorough examination of individual test documents and annotations returned by the evaluated system. These visualizations aid in identifying systematic problems with the annotator, ways to mitigate them (Figure 2), and help experts in better understanding why the evaluated system performed in a certain way.

5. Interpretability and Explainability in Orbis

This sections presents an overview of the interpretability and explainability features built into Orbis.

5.1. FAIR Principles

One key aspect in the evaluation of benchmarking tools is represented by the implementation of FAIR principles [42]. Out of the tools examined in Section 2, only GERBIL has previously been described as adhering to these principles [30]. Orbis also respects these principles, though the implementation in some cases differs from GERBIL. Table 3 summarizes the current Orbis implementation of the FAIR principles. In contrast to other benchmarking systems which mostly focus on reproducibility, the development of Orbis was also guided by the desire to provide potent means for understanding evaluation results which in turn should pave the way for a more systematic improvement of NEL annotators.

The main challenge to the issues of explainability and interpretability was to provide a set of possible interpretation on top of current results without resorting to explaining each algorithm. This meant that several compromises had to be made and that a new approach towards explainability and interpretability through visualization had to be developed. The following subsections describe how systematic error analysis, lenses and visual classification help in achieving this goal.

5.2. Error Analysis

As already outlined in Section 3.3, when performing NEL evaluations we can discover errors caused by the dataset (DS), by the KG version (KG), by the evaluated system (AN) or even by the scorer (SE) that was used. It has to be noted that not all of these error types can necessarily be automatically detected, as for example a system error might need multiple rounds of analysis (e.g., besides flagging such an error, a system designer might also want to know which component of the system caused the error). Some dataset errors like inconsistencies related to *contained* ($\mathcal{C}$) or *overlapping matches* ($\mathcal{P}$) can be spotted if annotation guidelines are known or at least if it is assumed that a single annotation style should be followed through an entire corpora annotation process. NIL classification errors, a special type of KG errors, can automatically be flagged when using a new KG version (e.g., when a tool that uses the latest DBpedia is evaluated against a gold standard produced with an earlier DBpedia version such as 3.9 and 2015-10). Unlike the other error cases, scorer errors are difficult to spot automatically, as it is often only due to the existence of a good scorer that we can flag any errors in the first place. Such automated error highlighting, even if not yet available for scorers, is definitely welcome for NEL system designers, as they will then be able to focus on the class of errors they are most interested in: annotator errors.

We need to note that a certain number of errors is caused by the evaluation setup. For example, evaluations might draw upon old gold standards or use different KG indexes (e.g., different KG versions). In our opinion, fair evaluations should include the same KG version in the gold standard and KG index used by the annotators. This can be achieved by using custom KG builds or annotation styles, as explained in the following subsection.

In addition, a case should be made for including the annotator settings with published evaluation results, since most annotator tools offer comprehensive configuration and fine-tuning options which are required for reproducing the reported outcome.

5.3. Lenses

Lenses enable transformations between gold standard and annotator results. Similar to their use in photography, they provide different views on gold standard annotations and, therefore, allow benchmarking across annotation styles and KB versions.

5.3.1. Annotation Styles

Problem description: Annotation styles specify rules that help both human and automated annotators in assessing (i) if candidate mentions should be considered a mention of a named entity, and (ii) the extent of the corresponding surface form.

Although a trivial design decision for isolated mentions, the consistent handling of nested mentions requires more thought. For instance, the text snippet *University of Western Australia Cricket Club* may contain, dependent on the applied annotation rule, up to four overlapping mentions (*Australia*, *Western Australia*, *University of Western Australia*, *University of Western Australia Cricket Club*). In addition, annotation styles might be entity type specific even within a single corpus.

Lens transformation rules: We consider the following three annotation styles, as illustrated based on the annotation of the text snippet *Vienna, VA*:

1. *ØMIN* disregards overlapping entities and extracts the minimum number of entities from the provided text: $m^{dbr:Vienna,\_Virginia}_{[\text{Vienna, VA}]}$, i.e., links the snippet to the *Vienna, Virginia* DBpedia entity.

Table 3
FAIR principles implementation in Orbis.

Principle	Implementation
Findable	
F1. (Meta)data are assigned a globally unique and persistent identifier	Unique ID per experiment
F2. Data are described with rich metadata (defined by R1 below)	Experimental configuration as YAML
F3. Metadata clearly and explicitly include the identifier of the data they describe	Data URIs are included
F4. (Meta)data are registered or indexed in a searchable resource	Planned for future version of Orbis
Accesible	
A1. (Meta)data are retrievable by their identifier using a standardised communications protocol	HTTP, JSON
A1.1 The protocol is open, free, and universally implementable	HTTP and JSON are open standards
A1.2 The protocol allows for an authentication and authorisation procedure, where necessary	Not needed
A2. Metadata are accessible, even when the data are no longer available	All experiments and data are archived.
Interoperable	
I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.	JSON
I2. (Meta)data use vocabularies that follow FAIR principles	Community-based, open vocabularies
I3. (Meta)data include qualified references to other (meta)data	Planned for future version of Orbis
Reusable	
R1. Meta(data) are richly described with a plurality of accurate and relevant attributes	Experiment measures are based on TAC standards.
R1.1. (Meta)data are released with a clear and accessible data usage license	GPL2
R1.2. (Meta)data are associated with detailed provenance	Provenance data can be added to all experiments.
R1.3. (Meta)data meet domain-relevant community standards	Covers a superset of domain-relevant data.

2. The annotation style *ØMAX*, in contrast, extracts the maximum number of entities (including overlaps): $m^{dbr:Vienna,\_Virginia}_{[\text{Vienna}]}, m^{dbr:Virginia}_{[\text{VA}]}$
3. The style *OMAX* essentially combines the previous styles while taking into account the overlaps: $m^{dbr:Vienna,\_Virginia}_{[\text{Vienna, VA}]}, m^{dbr:Virginia}_{[\text{VA}]}$

The presented rules only consider borderline cases, even though combinations of them can also be used within a corpus. For instance, a corpus might use the *OMAX* rule for LOC entities but apply *ØMIN* for all other entity types, therefore only yielding $m^{dbr:ETH\_Zurich}_{[\text{ETH Zurich}]}$ rather than $m^{dbr:ETH\_Zurich}_{[\text{ETH Zurich}]}$ and $m^{dbr:Zurich}_{[\text{Zurich}]}$ for the text snippet *ETH Zurich*.

Table 4 outlines transformation rules between different annotation styles. The table's header refers to the column of the source annotation style (i.e., ØMIN, ØMAX, OMAX) and the table rows indicate the destination style. A transformation from ØMAX to ØMIN, for instance, involves translating the non-overlapping mentions $m^{e_1,KG}_{[x1,y11]}, \ldots, m^{e_n,KG}_{[x1n,y1]}$ to a single mention $m^{e_1,KG}_{[x1,y1]}$. Applying this rule to the mention "Burbank, California" would transform the two ØMAX annotations

$$m^{dbr:Burbank,\_California}_{[\text{Burbank}]} \quad m^{dbr:California}_{[\text{California}]}$$

to the ØMIN annotation

$$m^{dbr:Burbank,\_California}_{[Burbank,California]}$$

that combines the surface forms of the ØMAX annotations. Table 5 provides additional examples that illustrate how these transformation rules translate between annotation styles.

5.3.2. Knowledge Graph Evolution and Translation

Problem description: Knowledge graphs such as DBpedia and Wikidata differ in their model, coverage and completeness. In addition, knowledge graphs evolve due to changes of the domain, corrections, and extensions of their coverage. Lenses can capture changes such as the

1. introduction of new entities;

Table 4
Lens transformation rules between the ØMIN, ØMAX and OMAX annotation styles.

Annotation style	ØMIN	ØMAX	OMAX
Corpus entity	$m^{e_1,KG}_{[x1,y1]}$	$m^{e_1,KG}_{[x1,y11]}, \ldots, m^{e_n,KG}_{[x1n,y1]}$	$m^{e_1,KG}_{[x1,y1]}, \ldots, m^{e_n,KG}_{[x1,y1]}$
Transformation to			
ØMIN	$m^{e_1,KG}_{[x1,y1]}$	$m^{e_1,KG}_{[x1,y1]}$	$m^{e_1,KG}_{[x1,y1]}$
ØMAX	$m^{e_1,KG}_{[x1,y11]}, \ldots, m^{e_n,KG}_{[x1n,y1]}$	$m^{e_1,KG}_{[x1,y11]}, \ldots, m^{e_n,KG}_{[x1n,y1]}$	$m^{e_1,KG}_{[x1,y11]}, \ldots, m^{e_n,KG}_{[x1n,y1]}$
OMAX	$m^{e_1,KG}_{[x1,y1]}, \ldots, m^{e_n,KG}_{[x1,y1]}$	$m^{e_1,KG}_{[x1,y1]}, \ldots, m^{e_n,KG}_{[x1,y1]}$	$m^{e_1,KG}_{[x1,y1]}, \ldots, m^{e_n,KG}_{[x1,y1]}$

Table 5
Examples of how annotation styles affect annotations based on the text snippets "Sir Patrik Steward OBE", "MLB Advanced Media (MLBAM)" and "Burbank, California".

Entity Type	**Person**	**Organization**	**Location**
Annotation Style			
ØMIN	Sir Patrick Stewart OBE	MLB Advanced Media (MLBAM)	Burbank, California
ØMAX	Sir Patrick Stewart OBE	MLB Advanced Media MLBAM	Burbank California
OMAX	Sir Patrick Stewart OBE Sir OBE	MLB Advanced Media (MLBAM) MLBAM	Burbank, California California

2. deletion of entities no longer considered relevant;
3. mapping changes like the addition of new properties or new entity types.

Lens transformation rules and mitigation strategies: Table 6 explains the corresponding transformation rules. New entities may allow the grounding of former *NIL* entities to the extended KG. The removal of an entity, in contrast, may transform an existing grounding to a *NIL* entity since the corresponding KG entity is no longer available. Finally, mapping changes may be geared towards creating a more fine-grained ontology and may lead to the introduction of new entity types.

KG translation (also known as KG migration) is the case in which a corpus that has been initially annotated with one KG is used to evaluate a component that links mentions to another KG. Many well maintained knowledge graphs such as DBpedia, GeoNames and Wikidata contain links to indicate equivalent entities (e.g., *owl:sameAs*, *skos:exactMatch*, etc). These links and techniques such as ontology alignment may be used to automatize the transformation of a mentions $m^{e_i,KG}_{[x_i,y_i]}$ within a KG (KG) to the corresponding mention $m^{e_i,KG'}_{[x_i,y_i]}$ in the target KG (KG'). KG migration draws at the same set of transformation rules as the KG evolution use case.

5.4. Visual Classification

Different visual classification schemas support users in performing a wide range of analyses. These schemes are paired with coloring schemes which help users in quickly identifying problematic results. Currently, Orbis supports selection of the following classification schemes:

- *Entity* - each distinct entity is presented with a different color. When examined across both panels (*gold* and *predicted*) the entities displayed with matching colors were correctly identified.
- *Type* - entities are classified by type. The coloring represents the typing in both panels.

Table 6
Lens transformation rules for knowledge graph evolution and translation.

Task	new entity	deleted entity	more fine grained entity mapping	coarser entity mapping
Corpus entity	$m^{nil,KG}_{[x_i,y_i]}$	$m^{e_i,KG}_{[x_i,y_i]}$	$m^{e_i,KG}_{[x_i,y_i]}$	$m^{e_{i1}}_{[x_{i1},y_{i1}]},\ldots,m^{e_{in},KG}_{[x_{in},y_{in}]}$
Transformation	$m^{e_i,KG'}_{[x_i,y_i]}$	$m^{nil,KG'}_{[x_i,y_i]}$	$m^{e_{i1},KG'}_{[x_{i1},y_{i1}]},\ldots,m^{e_{in},KG'}_{[x_{in},y_{in}]}$	$m^{e_i,KG'}_{[x_i,y_i]}$

– *Result* - the classification and coloring scheme reflects the test results (e.g., TP, FP, TN, FN). This coloring mostly affects the *predicted* panel.

Advanced classification schemes have also been developed for various plugins, but they typically have more dependencies (e.g., evaluation types, taxonomies or ontologies, corpora) and are enabled only for special evaluation types (e.g., slot filling). Some examples of such classifications include:

– *Cluster* - which is used for slot filling evaluations. All attributes that belong to the same entity are included in the same cluster (Figure 3). For example, if a text contains information about a *company*, its *CEO* and its *address* - they will all be included in the same company cluster and colored with the same color.
– The *Error* classification schema displays classification errors according to a taxonomy or ontology. The default taxonomy for Orbis is currently based on [5] in which each error class is assigned a different color
– *Content Extraction* - which supports evaluating the performance of content extraction pipelines (Figure 4). This is currently the only view that focuses on text blocks rather than entities, since it highlights whether the correct text snippets were extracted.

Orbis also provides an *overview* which offers additional information related to an evaluation (e.g., dataset, evaluation type). For debugging reasons, a reduced set of this functionality (e.g., only general settings and results) can be displayed on each page.

To enable reproducibility of old evaluations, Orbis can also serialize the whole content of the evaluation (e.g., corpora, gold and predicted results, general evaluation results, etc) in its custom archive format called *rucksack*. The created archives allow the reconstruction of the gold standard and annotator results at evaluation time, therefore, effectively providing time capsules with past results. In the case of copyright protected corpora or gold standards, we recommend the full encryption of such rucksack files to protect the contained data from leaking.

5.5. Comparison with Other Tools

The following discussion classifies benchmarking systems based on the following criteria:

– (i) the extent to which they provide explanations of evaluation results;
– (ii) how they aid experts in performing drill-down analyses of the evaluation results;
– (iii) the support they provide for various types of visualizations.

Leaderboard style only tools (e.g., KILT [28]) were not included in our comparison, since we are interested in systems that can be used as springboards for more complex evaluations.

As it can be easily seen from Tables 7 and 8, the tools that are popular for evaluating content extraction (e.g., Cleaneval and Waddle), do not offer advanced visualization facilities. This can be explained through the fact that they were not developed for specific use cases such as cleaning social media and web forums, but rather for the general task of cleaning web pages. Nevertheless, we decided to include them in our tables, since they are the most popular in their category. Some sub-tasks like boilerplate detection also have specific evaluation tools (e.g., CleanPortalEval[10]), which haven't been included, since they only cover a small part of the content extraction use case.

Tools should not be limited to aggregated evaluation metrics such as precision, recall, F1, and the counts of false positives and false negatives within a document, but also explain results by providing annotations such as `incorrect surface form`, and `incorrect link`, if possible. These descriptions are termed primary analysis since they provide a basic explanation

[10]https://github.com/ppke-nlpg/CleanPortalEval

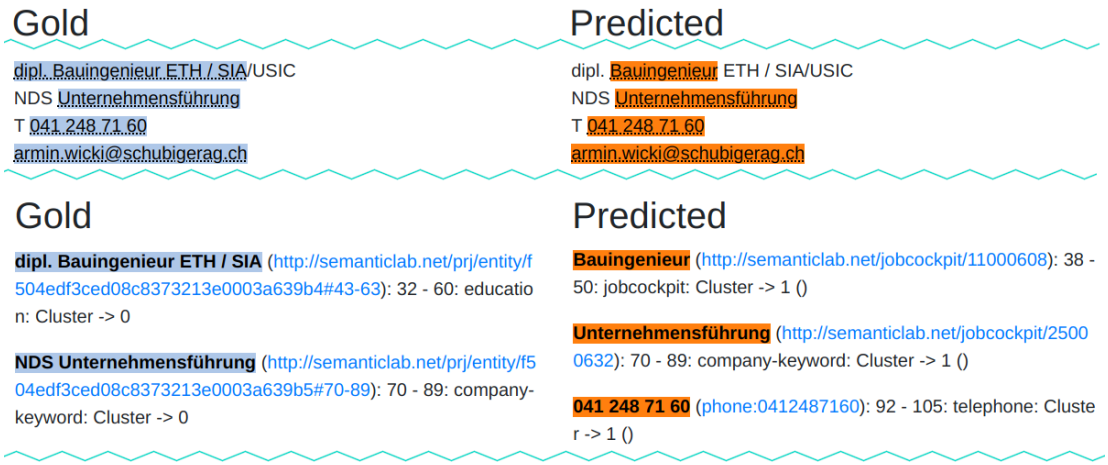

Fig. 3. A cropped screenshot of cluster classification scheme for the JobCockpit corpus. The zigzag lines indicate the removal of white lines. Task: Slot Filling.

Fig. 4. A cropped screenshot of the Orbis - Harvest results for the WebForum52 corpus. Task: Forum Extraction

Table 7
Comparison of results provided by different CE/NEL/Combined evaluation tools.

		Result granularity				Analysis			Export
	type	corpus	doc	sentence	NE	primary	fine-grained	third-party	formats
Cleaneval	CE	✓	✓	✗	✗	✗	✗	✗	✗
Waddle	CE	✓	✓	✗	✗	✗	✗	✗	✗
BAT	NEL	✓	✓	✓	✓	✗	✗	✗	✗
GERBIL	NEL	✓	✓	✓	✓	✗	✗	✓	QB
neleval	NEL	✓	✓	✓	✓	✓	✗	✓	tsv
VEX	NEL	✓	✓	✓	✓	✓	✗	✗	✗
Orbis	CE/NEL/SF	✓	✓	✓	✓	✓	✓	✓	csv, tsv, json, html

Table 8
Comparison of the support for drill-down analysis provided by different CE/NEL/Combined evaluation tools.

		Visualizations & drill-down analysis						
	type	doc	sentence	NE	Δ between runs	single visualization	parallel visualizations	KG lookup
Cleaneval	CE	✗	✗	✗	✗	none	✗	✗
Waddle	CE	✗	✗	✗	✗	none	✗	✗
BAT	NEL	✗	✗	✗	✗	measurements	✗	✗
GERBIL	NEL	✗	✗	✗	✗	measurements	✗	✗
neleval	NEL	✗	✗	✗	✗	✗	✗	✗
VEX	NEL	✓	✓	✓	✗	gold-annotator	✗	✗
Orbis	CE/NEL/SF	✓	✓	✓	corpus, document	gold-annotator	2+ gold	✓

of the classification results. More fine-grained explanations complement the primary analysis by clarifying aspects related to the origin of the errors (e.g., in the KG, corpora or annotator).

The analysis of NEL evaluation tools in Table 7 considers systems that act as black boxes as being limited to returning aggregated results. Once a system provides primary analysis results, it is considered to belong to the explainable systems category, even if their support for explainability is minimal. The first four columns (corpus, document, sentence, NE) describe the granularity of the results provided by the evaluation systems. As it can be seen, all the systems compute granular results, even though they might not all be accessible to developers. The next set of columns describes the error analysis capabilities. In some cases, it is also possible to provide a fine-grained analysis with an external tool or script, as described in [5, 18]. The last column lists the output formats supported by the investigated NEL evaluation tools.

Another important aspect is the interpretability and explainability of results in terms of the visualization capabilities offered by a tool. Most benchmarking tools that could be classified as explainable can indeed provide visualization capabilities, but we have deemed such capabilities insufficient if they are not paired with fine-grained analysis of the results. Visual capabilities of the analyzed systems are presented in Table 8. We have examined their ability to provide details on the difference between runs, as well as single or parallel visualization and KG lookup capabilities (e.g., the ability to show additional details about the entity referenced by a certain URI).

6. Discussion

Developing Orbis has helped us in improving our own tools, development and evaluation processes and has considerably shaped the way we approach development of new software. This chapter explains both its impact on our own tools, as well as suggestions on how others can use Orbis to improve their IE pipelines.

6.1. Impact

Orbis was developed while extending our ideas on error categories observed in NEL tools developed in the InVID[5] and DISCOVER [39] research projects. We realized that in some cases it was difficult for developers to understand where various errors come from, even when they were clearly labelled, since they were in text files generated by command line tools which did not provide the necessary context for a deeper analysis. These problems considerably impeded our NEL research efforts and triggered the decision to create Orbis, a system that supports explainable benchmarking and evaluations through a visual interface. The first step was to create a general architecture upon which to build the system [26]. The initial Orbis version included a general pipeline for processing information, a basic NIF reader, a plugin mechanism, a scorer and the early visual interface. Later versions added lenses, new evaluation types, multiple evaluation domains and many significant improvements to the interface. The early adoption of Orbis resulted in improved deployments of our NEL annotator (Recognyze) for a variety of national and European projects (e.g., financed through HORIZON 2020, FFG, Innossuise) including ReTV, EPOCH [11], DISCOVER, Job-Cockpit, MedMon [12] and CareerCoach.

Although Orbis started with a sole focus on named entities extraction, its success motivated the extension to other research areas. The first new use case ap-

[11] https://epoch-project.eu
[12] https://www.fhgr.ch/medmon

peared as an extension of our expertise in the area of NEL and yielded a small tool for showcasing slot filling results. The idea was to simply highlight the entities that belong to various slots or clusters. This tool was developed in Java and JavaScript in parallel with the first version of Orbis. The JavaScript visualization code was later merged into Orbis.

A second use case appeared a year later, when we noticed that many forum extraction tools have been prone to similar extraction errors. These errors included splitting forum posts into multiple texts, removing post authors or dates, or even deleting entire posts. This was far from ideal, and we required a better method for understanding what happened to the missing data or posts before addressing this problem. Displaying extraction errors visually helped a lot, and triggered the development of Harvest [37] which ended up being one of the top tools in the field of forum extraction. Harvest has been successfully deployed on various projects that use the webLyzard platform [13], including some associated portals.

6.2. Improving Information Extraction Tools with Orbis

There is no perfect algorithm that performs best on all datasets. The variation between results obtained with Orbis and other tools is due to the strict entity typing policy. Lenses offer a solution to this problem, but they require careful calibration when implemented, as it is important to strike the right balance between the number of changes we need to explore and the number of lenses through which we can implement these changes. A smaller number of lenses is preferred to a higher number, as nobody has enough time to examine too many views.

Besides looking at the general results, each document can be analyzed through multiple classification schemes, as discussed in Section 4.3. Each classifier serves different purposes, as the *entity* (default) scheme helps people identify the various entities classified; the *type* scheme highlights the various types from a text; whereas the *result* classification is ideal for understanding what kind of test results are more frequent in certain documents. Based on these classifications, developers can select documents or error types they want to improve. Developers may, for example, opt to simply fix all the errors that can be discovered in the basic entity types (hence they might use the *Type* classification), whereas others might focus on addressing all the errors (hence they will prefer the test *Results* classification).

Orbis can also be used for improving gold standards. Its visualizations and the results that can be extracted from them represent a first step. For example, after selecting a classification scheme and verifying the results, one developer might submit a set of changes to the publisher of the corpora. This is relatively easy, as all the results of the evaluations (e.g., including the results of the various classification schemes) are available in an easy to parse JSON format.

Different annotation styles, in contrast, can be addressed by using lenses. The simple fact that a NEL tool can lean heavily towards one or more annotation styles means that it might be severely penalized when the gold standard was created using a different annotation style. Lenses are capable of bridging these differences and, therefore, provide much more nuanced results.

KG evolution is another serious issue which can be addressed with Orbis lenses since they otherwise have a significant impact on the evaluation result. Experiments suggest that researchers who change annotation styles or scoring rules need to properly document these changes and to publish the refined gold standard together with the evaluation results in order to ensure reproducibility. If the evaluation considers multiple lenses, researchers will be rewarded with more comprehensive results that also reflect how the annotators perform in different settings.

7. Conclusion

This paper described Orbis, an explainable framework for performing evaluations of information extraction tasks such as content extraction, named entity recognition, named entity linking and slot filling. Orbis visualizes evaluation results enabling system developers to quickly inspect and better understand errors. In contrast to other evaluation tools which often only consider one evaluation task and are frequently limited to aggregated metrics or very rudimentary information on linking errors, Orbis supports multiple tasks along the information extraction pipeline, displays gold standard and the annotators output within their textual context, provides information on all linked entities and means to obtain further background information and error analysis on these entities. As opposed to clas-

[13] www.weblyzard.com

sic content extraction tools which used visualization to showcase the text-rich regions, for example, Orbis also provides visualizations for the extracted content, allowing developers to understand where mistakes may have originated.

The visualizations allow researchers to quickly compare the performance of two systems with each other and the gold standard. This evaluation mode is particularly useful in assessing the effects of architectural changes and in evaluating the strengths and weaknesses of different IE systems.

As outlined through the presented use cases, Orbis significantly lowers the effort required to perform drill-down analysis which in turn enable researchers to locate a problem in algorithms, machine learning components, gold standards and data sources more quickly, leading to a more efficient allocation of research efforts and developer resources.

Since Orbis is a flexible framework and offers an affordable option for building new evaluation use cases, it can easily be argued that it is in fact a framework designed to help build evaluation infrastructure. The focus on multiple steps of the IE pipeline, and the support for lenses and on understanding the variation in results caused by the different annotation styles, KGs or entity classification schemes differentiates Orbis from the rest of the available benchmarking tools.

Future work will be focused on (i) integrating statistical significance tests such as the Wilcoxon Rank Sum test into the Orbis platform; (ii) creating plugins for tracking and publishing evaluation results; (iii) developing support for additional evaluation types, as well as for other tasks like sentiment analysis, relation extraction; and (iv) developing Orbis into a full-fledged benchmarking solution that supports the whole evaluation process, from creating annotations and lenses to reproducible workflows and visual error analysis.

Acknowledgements

This work has been supported through the CareerCoach project (https://www.fhgr.ch/careercoach) funded by Innosuisse; the ReTV project (https://retv-project.eu/) funded through the European Union's Horizon 2020 Research and Innovation Programme under GA No 780656 and FFG project EPOCH (htthttps://epoch-project.eups://epoch-project.eu).

The authors would like to thank Dr. Giuseppe Rizzo, Philipp Kuntschik, Alexander van Schie and Corsin Capol for contributing to the earlier versions of code or ideas presented in this work.

References

[1] Adel, H., 2018. Deep Learning Methods for Knowledge Base Population. Ph.D. thesis. Ludwig–Maximilians–Universität München. Munich, Germany.

[2] Al-Moslmi, T., Ocaña, M.G., Opdahl, A.L., Veres, C., 2020. Named entity extraction for knowledge graphs: A literature overview. IEEE Access 8, 32862–32881. URL: https://doi.org/10.1109/ACCESS.2020.2973928, doi:.

[3] Barbaresi, A., Lejeune, G., 2020. Out-of-the-box and into the ditch? multilingual evaluation of generic text extraction tools, in: Barbaresi, A., Bildhauer, F., Schäfer, R., Stemle, E. (Eds.), Proceedings of the 12th Web as Corpus Workshop, WAC@LREC 2020, Marseille, France, May 2020, European Language Resources Association. pp. 5–13. URL: https://www.aclweb.org/anthology/2020.wac-1.2/.

[4] Baroni, M., Chantree, F., Kilgarriff, A., Sharoff, S., 2008. Cleaneval: a competition for cleaning web pages, in: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), European Language Resources Association (ELRA), Marrakech, Morocco. URL: http://www.lrec-conf.org/proceedings/lrec2008/pdf/162_paper.pdf.

[5] Braşoveanu, A.M.P., Rizzo, G., Kuntschick, P., Weichselbraun, A., Nixon, L.J., 2018. Framing named entity linking error types, in: Calzolari, N., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., Tokunaga, T. (Eds.), Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Paris, France. pp. 266–271. URL: http://www.lrec-conf.org/proceedings/lrec2018/summaries/612.html.

[6] Bunescu, R.C., Pasca, M., 2006. Using encyclopedic knowledge for named entity disambiguation, in: McCarthy, D., Wintner, S. (Eds.), EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy, The Association for Computer Linguistics. pp. 9–16. URL: http://aclweb.org/anthology/E/E06/E06-1002.pdf.

[7] Choi, J.D., Tetreault, J.R., Stent, A., 2015. It depends: Dependency parser comparison using A web-based evaluation tool, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, The Association for Computer Linguistics. pp. 387–396. URL: https://doi.org/10.3115/v1/p15-1038, doi:.

[8] organization committee, T., 2017. Cold Start Knowledge Base Population at TAC 2017 - Task Description, in: Proceedings of the 10th Text Analysis Conference (TAC 2017)., NIST, Gaithersburg, Maryland, USA. p. 1. URL: https://tac.nist.gov/2017/KBP/ColdStart/guidelines/TAC_KBP_2017_ColdStartTaskDescription_1.0.pdf.

[9] Cornolti, M., Ferragina, P., Ciaramita, M., 2013. A framework for benchmarking entity-annotation systems, in: 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, International World Wide Web Conferences Steering Committee / ACM. pp. 249–260. URL: http://dl.acm.org/citation.cfm?id=2488411.

[10] Daiber, J., Jakob, M., Hokamp, C., Mendes, P.N., 2013. Improving efficiency and accuracy in multilingual entity extraction, in: I-SEMANTICS 2013 - 9th International Conference on Semantic Systems, ISEM '13, Graz, Austria, September 4-6, 2013, ACM. pp. 121–124. URL: http://doi.acm.org/10.1145/2506182.2506198, doi:.

[11] Delpeuch, A., 2019. Opentapioca: Lightweight entity linking for wikidata. CoRR abs/1904.09131. URL: http://arxiv.org/abs/1904.09131, arXiv:1904.09131.

[12] Dojchinovski, M., Sasaki, F., Gornostaja, T., Hellmann, S., Mannens, E., Salliau, F., Osella, M., Ritchie, P., Stoitsis, G., Koidl, K., Ackermann, M., Chakraborty, N., 2016. FREME: multilingual semantic enrichment with linked data and language technologies, in: Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (Eds.), Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016., European Language Resources Association (ELRA). pp. 4180–4183. URL: http://www.lrec-conf.org/proceedings/lrec2016/summaries/578.html.

[13] Hachey, B., Nothman, J., Radford, W., 2014. Cheap and easy entity evaluation, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers, The Association for Computer Linguistics. pp. 464–469. URL: http://aclweb.org/anthology/P/P14/P14-2076.pdf.

[14] Heer, J., Shneiderman, B., 2012. Interactive Dynamics for Visual Analysis. Communications of the ACM 55, 45–54. URL: https://dl.acm.org/citation.cfm?id=2133821, doi:.

[15] Heinzerling, B., Strube, M., 2015. Visual error analysis for entity linking, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, System Demonstrations, ACL. pp. 37–42. URL: http://aclweb.org/anthology/P/P15/P15-4007.pdf.

[16] Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G., 2011. Robust disambiguation of named entities in text, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 782–792. URL: http://www.aclweb.org/anthology/D11-1072.

[17] Ilievski, F., Vossen, P., Schlobach, S., 2018. Systematic study of long tail phenomena in entity linking, in: Bender, E.M., Derczynski, L., Isabelle, P. (Eds.), Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018, Association for Computational Linguistics. pp. 664–674. URL: https://aclanthology.info/papers/C18-1056/c18-1056.

[18] Jha, K., Röder, M., Ngomo, A.N., 2017. All that glitters is not gold - rule-based curation of reference datasets for named entity recognition and entity linking, in: Blomqvist, E., Maynard, D., Gangemi, A., Hoekstra, R., Hitzler, P., Hartig, O. (Eds.), The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I, pp. 305–320. URL: https://doi.org/10.1007/978-3-319-58068-5_19, doi:.

[19] Ji, H., Pan, X., Zhang, B., Nothman, J., Mayfield, J., McNamee, P., Costello, C., 2017. Overview of TAC-KBP2017 13 languages entity discovery and linking, in: Proceedings of the 2017 Text Analysis Conference, TAC 2017, Gaithersburg, Maryland, USA, November 13-14, 2017, NIST. p. 4. URL: https://tac.nist.gov/publications/2017/papers.html.

[20] Ji, H., Sil, A., Dang, H.T., Soboroff, I., Nothman, J., 2019. Overview of TAC-KBP2019 Fine-grained Entity Extraction, in: Proceedings of the 12th Text Analysis Conference (TAC 2019), p. 1.

[21] Kohlschütter, C., Fankhauser, P., Nejdl, W., 2010. Boilerplate detection using shallow text features, in: Davison, B.D., Suel, T., Craswell, N., Liu, B. (Eds.), Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010, ACM. pp. 441–450. URL: https://doi.org/10.1145/1718487.1718542, doi:.

[22] Ling, X., Singh, S., Weld, D.S., 2015. Design challenges for entity linking. TACL 3, 315–328. URL: https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/528.

[23] Moro, A., Navigli, R., 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking, in: Cer, D.M., Jurgens, D., Nakov, P., Zesch, T. (Eds.), Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015, The Association for Computer Linguistics. pp. 288–297. URL: http://aclweb.org/anthology/S/S15/S15-2049.pdf.

[24] Moro, A., Raganato, A., Navigli, R., 2014. Entity linking meets word sense disambiguation: a unified approach. Transactions of the Association for Computational Linguistics 2, 231–244.

[25] Nuzzolese, A.G., Gentile, A.L., Presutti, V., Gangemi, A., Meusel, R., Paulheim, H., 2016. The second open knowledge extraction challenge, in: Sack, H., Dietze, S., Tordai, A., Lange, C. (Eds.), Semantic Web Challenges - Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers, Springer. pp. 3–16. URL: https://doi.org/10.1007/978-3-319-46565-4_1, doi:.

[26] Odoni, F., Kuntschik, P., Braşoveanu, A.M.P., Weichselbraun, A., 2018. On the importance of drill-down analysis for assessing gold standards and named entity linking performance, in: Fensel, A., de Boer, V., Pellegrini, T., Kiesling, E., Haslhofer, B., Hollink, L., Schindler, A. (Eds.), Proceedings of the 14th International Conference on Semantic Systems, SEMANTICS 2018, Vienna, Austria, September 10-13, 2018, Elsevier. pp. 33–42. URL: https://doi.org/10.1016/j.procs.2018.09.004, doi:.

[27] Peters, M.E., Lecocq, D., 2013. Content extraction using diverse feature sets, in: Carr, L., Laender, A.H.F., Lóscio, B.F., King, I., Fontoura, M., Vrandecic, D., Aroyo, L., de Oliveira, J.P.M., Lima, F., Wilde, E. (Eds.), 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume, International World Wide Web Conferences Steering Committee / ACM. pp. 89–90. URL: https://doi.org/10.1145/2487788.2487828, doi:.

[28] Petroni, F., Piktus, A., Fan, A., Lewis, P.S.H., Yazdani, M., Cao, N.D., Thorne, J., Jernite, Y., Karpukhin, V., Maillard, J., Plachouras, V., Rocktäschel, T., Riedel, S., 2021. KILT: a benchmark for knowledge intensive language tasks, in: Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., Zhou, Y. (Eds.), Proceedings of the 2021 Conference of the

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, Association for Computational Linguistics. pp. 2523–2544. URL: https://doi.org/10.18653/v1/2021.naacl-main.200, doi:.

[29] Pomikálek, J., 2011. jusText. URL: http://hdl.handle.net/11858/00-097C-0000-000D-F696-9. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[30] Röder, M., Usbeck, R., Ngomo, A.N., 2018. GERBIL - benchmarking named entity recognition and linking consistently. Semantic Web 9, 605–625. URL: https://doi.org/10.3233/SW-170286, doi:.

[31] Rosales-Méndez, H., Hogan, A., Poblete, B., 2019. Nifify: Towards better quality entity linking datasets, in: Amer-Yahia, S., Mahdian, M., Goel, A., Houben, G., Lerman, K., McAuley, J.J., Baeza-Yates, R.A., Zia, L. (Eds.), Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019., ACM. pp. 815–818. URL: https://doi.org/10.1145/3308560.3316465, doi:.

[32] Rosales-Méndez, H., Poblete, B., Hogan, A., 2018. What should entity linking link?, in: Olteanu, D., Poblete, B. (Eds.), Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management, Cali, Colombia, May 21-25, 2018., CEUR-WS.org. p. 15. URL: http://ceur-ws.org/Vol-2100/paper10.pdf.

[33] Shneiderman, B., 1996. The eyes have it: A task by data type taxonomy for information visualizations, in: Proceedings of the IEEE Symposium on Visual Languages, IEEE VL 1996, IEEE. pp. 336–343. URL: https://www.cs.umd.edu/~ben/papers/Shneiderman1996eyes.pdf, doi:.

[34] Sundheim, B., 1995. Overview of results of the MUC-6 evaluation, in: Proceedings of the 6th Conference on Message Understanding, MUC 1995, Columbia, Maryland, USA, November 6-8, 1995, ACL. pp. 13–31. URL: https://doi.org/10.3115/1072399.1072402, doi:.

[35] Waitelonis, J., Jürges, H., Sack, H., 2016. Don't compare apples to oranges: Extending GERBIL for a fine grained NEL evaluation, in: Fensel, A., Zaveri, A., Hellmann, S., Pellegrini, T. (Eds.), Proceedings of the 12th International Conference on Semantic Systems, SEMANTICS 2016, Leipzig, Germany, September 12-15, 2016, ACM. pp. 65–72. URL: https://doi.org/10.1145/2993318.2993334, doi:.

[36] Waitelonis, J., Jürges, H., Sack, H., 2019. Remixing entity linking evaluation datasets for focused benchmarking. Semantic Web 10, 385–412. URL: https://doi.org/10.3233/SW-180334, doi:.

[37] Weichselbraun, A., Brasoveanu, A.M., Waldvogel, R., Odoni, F., 2020a. Harvest - an open source toolkit for extracting posts and post metadata from web forums, in: IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2020), pp. 1–7. doi:. accepted 27 October 2020.

[38] Weichselbraun, A., Kuntschik, P., Braşoveanu, A.M.P., 2018. Mining and leveraging background knowledge for improving named entity linking, in: Proceedins of the 8th International Conference on Web Intelligence, Mining and Semantics (WIMS 2018), ACM, Novi Sad, Serbia. pp. 27:1–27:11. URL: http://doi.acm.org/10.1145/3227609.3227670, doi:.

[39] Weichselbraun, A., Kuntschik, P., Hörler, S., 2020b. Improving company valuations with automated knowledge discovery, extraction and fusion. Information - Wissenschaft und Praxis 71, 1–5. arXiv:https://arxiv.org/abs/2010.09249.

[40] Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Saiz, L.A., Cardoso, M.J., Maier-Hein, L., Kopp-Schneider, A., 2021. Methods and open-source toolkit for analyzing and visualizing challenge results. Scientific Reports 11, 1–15.

[41] Wilkinson, L., 2005. The Grammar of Graphics (Statistics and Computing). Statistics and Computing, Springer-Verlag New York, Secaucus, NJ, USA. URL: https://www.springer.com/de/book/9780387245447, doi:.

[42] Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.W., da Silva Santos, L.B., Bourne, P.E., et al., 2016. The fair guiding principles for scientific data management and stewardship. Scientific data 3.