ELEVANT: A Fully Automatic Fine-Grained Entity Linking Evaluation and Analysis Tool

Hannah Bast and **Matthias Hertel** and **Natalie Prange**
University of Freiburg
Freiburg, Germany
{bast,hertelm,prange}@cs.uni-freiburg.de

Abstract

We present *Elevant*, a tool for the fully automatic fine-grained evaluation of a set of entity linkers on a set of benchmarks. Elevant provides an automatic breakdown of the performance by various error categories and by entity type. Elevant also provides a rich and compact, yet very intuitive and self-explanatory visualization of the results of a linker on a benchmark in comparison to the ground truth. A live demo, the link to the complete code base on GitHub and a link to a demo video are provided under https://elevant.cs.uni-freiburg.de .

1 Introduction

Entity linking is a fundamental problem, and a first step for or component of many NLP applications. In this paper, we consider end-to-end entity linking systems, which do the following: given a text and a set of entities, identify each mention of one of those entities in the text and say which of these entities it is.

Due to its fundamental importance and wide applicability, there is vast literature on entity linking, and also a large number of concrete software tools. Many publications also come with an evaluation, which compares the entity linker introduced in the publication with existing linkers, usually on a variety of benchmarks. There are also several standard benchmarks, which are found in many evaluations. This is a positive and pleasing development.

The typical statistics include overall precision and recall, that is, which percentage of the found mentions were correct and which percentage of the correct mentions were found. It is a frequent experience that the numbers in the evaluation are very good, yet the experience when applying that entity linker in an own application are less convincing. And not so rarely, there is even trouble reproducing the results from the publication.

The problem is that overall precision, recall, and F1 tell us little about the particular strengths and weaknesses of a particular entity linker for a particular application. Particular benchmarks often require very particular skills from an entity linker, and an entity linker may be deliberately or unknowingly tuned towards these particularities. To find out about the strengths and weaknesses of an entity linker, one needs to look at the results in more detail, which typically has *three* aspects:

(1) look at particular types of errors,
(2) look at particular types of entities,
(3) look at particular pieces of text.

Doing this on the raw input and output files is tedious, so that often small scripts are written to aid this process. However, these scripts are usually quite basic and imperfect. It also means that researchers do the same work over and over again.

It is the purpose of this paper to provide a comprehensive and easy-to-use tool, which every entity-linking researcher can and wants to use to analyze and understand the performance of a particular entity linker in detail.

1.1 Contributions

We consider these as our main contributions:

• We provide *Elevant*, a tool for the fully automatic fine-grained evaluation of a given set of entity linkers on a given set of benchmarks. The evaluation has two parts: a table with overall and fine-grained statistics, and a panel that provides a rich visualization of the concrete results that contribute to a selected statistics from the table.

• The table provides one row per experiment (a particular entity linker evaluated on a particular benchmark). Each column stands for one of a rich set of error categories: all errors, various kinds of entity detection errors, various kinds of entity disambiguation errors, errors on entities of particular type. There are controls to show and hide individ-

Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 72 - 79

ual columns or groups of columns. A screenshot of this part of the tool is shown in Figure 1.

• For each table cell, Elevant provides a rich visualization of the result of that particular entity linker on that benchmark for that category. The visualization is compact and intuitive, providing for each text record all information about true positives, false positives, and false negatives. The information is displayed with intuitive highlights and more detailed information provided on mouseover. Figure 2 shows a screenshot of this visualization.

• As a result of the compact single-column visualization, Elevant can also provide an intuitive side-by-side comparison of the concrete results of two experiments.

• For each table column, Elevant can generate a graph that shows the results of all linkers over all benchmarks in the table for that category. An example for such a graph is shown in Figure 3.

• The code for Elevant is open source, well documented, and easy to use. We support various standard formats for both the benchmarks and the results from the entity linkers. There is a demo website available under `https://elevant.cs.uni-freiburg.de`, which shows the results of a variety of well-known entity linkers evaluated on a variety of well-known benchmarks.

2 Related Work

GERBIL (Usbeck et al., 2015) is similar to Elevant in that it provides a web-based platform for the comparison of a given set of entity linking systems on a given set of benchmarks. GERBIL is widely used and has helped standardizing benchmarks and the evaluation measures on these benchmarks. Elevant goes one step further by not only providing aggregate measures for the whole benchmarks (like precision, recall, and F1), but a detailed breakdown of the results by error category and entity type, and the ability to look at the results of different methods in detail, both in comparison to the ground truth and in comparison with each other.

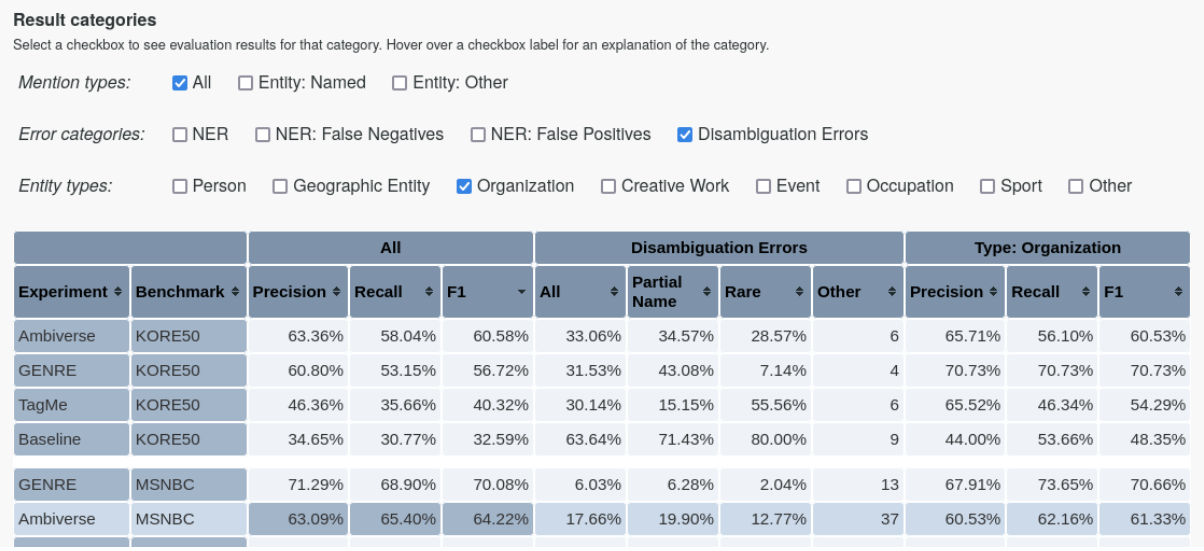
ORBIS (Odoni et al., 2018) is similar to Elevant in that it provides overall statistics and a visualization of individual annotations. However, ORBIS does not provide a fine-grained error analysis and the visualization is less rich and less compact compared to that of Elevant. In particular, the visualization uses two columns: one for the annotations from the entity linker and one for those from the ground truth. This makes it difficult to grasp the most important information at one glance and there is no support for the comparison of two entity linkers. While the source code is publicly available and easy to install, errors can occur during usage due to dependency issues. Elevant avoids this issue by providing an easy-to-use docker setup.

VEX (Heinzerling and Strube, 2015) is a web app for visual error analysis of entity linking systems. Benchmark texts are displayed with highlighted predicted entities and ground truth entities. The highlights are color-coded such that true positives, false positives and false negatives can be distinguished. VEX focuses on showing clusters of entities, that is, indicating which predicted mentions and ground truth mentions have been linked to the same entity. For this purpose, identical entities are connected via lines. VEX does not display a system's evaluation results and does not allow direct comparison of different systems.

As part of their work on entity linking on Wikipedia, Klang and Nugues (2018) provide a system for visualizing annotations in Wikipedia articles such as hyperlinks or an entity linking tool's entity predictions. Identical entities are visualized by using the same annotation color. The tool is not meant for evaluating linking results against a ground truth. That is, no ground truth entities are displayed and the tool does not provide information about true positives, false positives or false negatives.

Strobl et al. (2020) provide a similar but more rudimentary system as part of their work on entity linking on Wikipedia. Predicted links are shown as hyperlinks to Wikipedia articles which correspond to the predicted entity. The color of the hyperlink indicates whether the hyperlink is an original intra-Wikipedia hyperlink or has been added by the entity linking system. An additional color is used for predicted unknown entities.

Multiple publications propose a fine-grained evaluation of entity linking systems on different entity types or frequent linking errors. Ling and Weld (2012) and Gillick et al. (2014) assign fine-grained types to recognized entities. Ling et al. (2015) discuss difficult decisions in the design of entity linking systems and benchmarks, which are common sources of linking errors, such as whether to link common entities, how specific the entities should be, which entities to link in case of metonymies, considered entity types and overlapping entities.



Result categories

Select a checkbox to see evaluation results for that category. Hover over a checkbox label for an explanation of the category.

Mention types: ☑ All ☐ Entity: Named ☐ Entity: Other

Error categories: ☐ NER ☐ NER: False Negatives ☐ NER: False Positives ☑ Disambiguation Errors

Entity types: ☐ Person ☐ Geographic Entity ☑ Organization ☐ Creative Work ☐ Event ☐ Occupation ☐ Sport ☐ Other

		All			Disambiguation Errors				Type: Organization		
Experiment	Benchmark	Precision	Recall	F1	All	Partial Name	Rare	Other	Precision	Recall	F1
Ambiverse	KORE50	63.36%	58.04%	60.58%	33.06%	34.57%	28.57%	6	65.71%	56.10%	60.53%
GENRE	KORE50	60.80%	53.15%	56.72%	31.53%	43.08%	7.14%	4	70.73%	70.73%	70.73%
TagMe	KORE50	46.36%	35.66%	40.32%	30.14%	15.15%	55.56%	6	65.52%	46.34%	54.29%
Baseline	KORE50	34.65%	30.77%	32.59%	63.64%	71.43%	80.00%	9	44.00%	53.66%	48.35%
GENRE	MSNBC	71.29%	68.90%	70.08%	6.03%	6.28%	2.04%	13	67.91%	73.65%	70.66%
Ambiverse	MSNBC	63.09%	65.40%	64.22%	17.66%	19.90%	12.77%	37	60.53%	62.16%	61.33%

Figure 1: Evaluation results for various experiments. The types used in the per-type evaluation are configurable. The real web app contains many more error categories (see Section 4.1).

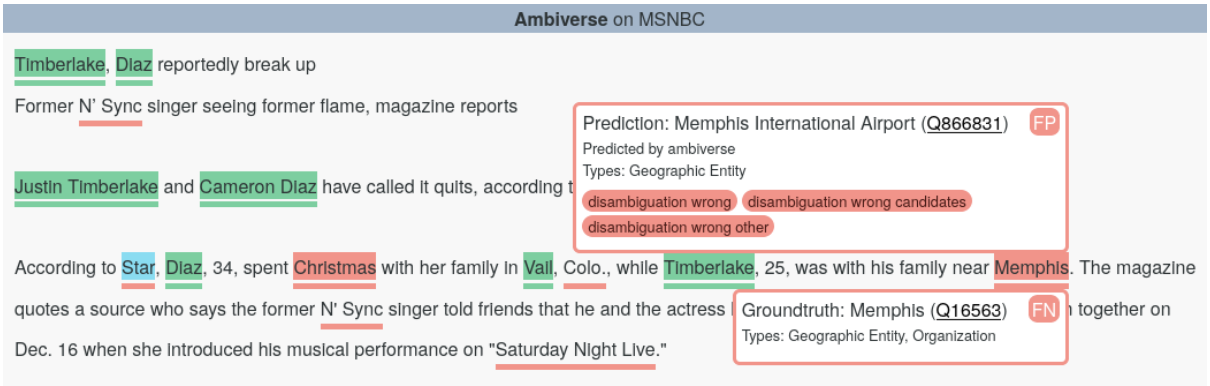


Figure 2: Visualization of predicted entities (highlighted text) and ground truth labels (underlined text) for a specific system (Ambiverse) and benchmark (MSNBC). True positives are shown in green, false positives and false negatives in red and unevaluated unknown entities in blue. Detailed information for each annotation is shown on mouseover.

Many of these decisions motivate our error types in Section 4.1. Rosales-Méndez et al. (2019) manually relabel three benchmarks to evaluate linking systems among dimensions such as the mention's base form, part of speech and overlap. Brasoveanu et al. (2018) propose an error classification based on the source of the error (e.g. knowledge base errors, dataset errors, annotator errors, etc.). They then manually categorize errors into these classes for a selected set of benchmarks and entity linking systems. Elevant follows this trend and goes one step further by providing a fully automatic classification into fine-grained error categories. Thus, by eliminating the need for expensive human labor, Elevant makes the fine-grained evaluation of entity linking errors feasible on a large scale.

3 Basic Principles

The core of Elevant is a web app that helps users analyze and compare results of various entity linking systems over various benchmarks in great detail. To this end, the user can add an experiment and evaluate its results using Elevant. We define an experiment as a run of a particular linker with particular settings over a particular benchmark. The pipeline for adding an experiment is as follows:

(1) add the benchmark (unless it already exists),
(2) run an entity linker on that benchmark,
(3) automatically evaluate the result in detail.

The following subsections explain how each of these steps can be executed using Elevant.

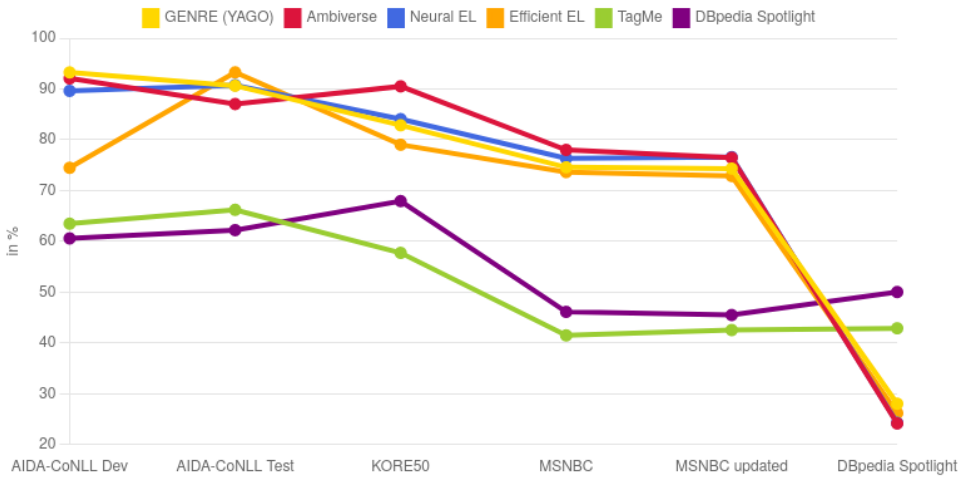


Figure 3: A graph generated by Elevant showing the NER F1 score for various entity linkers and benchmarks.

3.1 Adding a benchmark

In order to add a benchmark to Elevant, it is enough to run a single Python script. This script converts a given input benchmark into Elevant's internally used article file format. Files in this format contain one JSON object per line which holds information for a single article such as its text, title, ground truth labels or entity predictions. Additionally, the script annotates ground truth labels with the entities' types (for a fine-grained per-type evaluation) and the entities' names (for presentation purposes). Elevant supports three different benchmark formats: the common NLP Interchange Format (NIF), the IOB format used by Hoffart et al. (2011) for their AIDA-CoNLL dataset, and a simple JSONL format that only requires information about the benchmark's article texts, the ground truth label spans and the corresponding references to ground truth entities. Entity references can be from Wikidata, Wikipedia or DBpedia. Entity references from Wikipedia or DBpedia are internally converted to Wikidata. Several popular entity linking benchmarks are already included in Elevant (see Section 4.7) and can be used out of the box.

3.2 Running an entity linker

In order to produce entity linking results that can be evaluated with Elevant, the user has two options:

(1) They can feed the output of the entity linker they wish to evaluate into a provided Python script that converts the linking results into Elevant's internal format. The script supports linking results in NIF, the Ambiverse (Hoffart et al., 2011) output format or a simple JSONL format that only requires information about the predicted entity spans and corresponding entity references. Like ground truth entity references, references to predicted entities can be from Wikidata, Wikipedia or DBpedia and are converted to Wikidata internally.

(2) They can implement the entity linker within Elevant. The same Python script used for converting entity linking results can then be used to produce new linking results in the required format with the implemented linker. Several entity linkers are already implemented (detailed in Section 4.8) and can be used out of the box.

3.3 Evaluating entity linking results

Once the entity linking results are in the required format, they can be evaluated with another Python script. That script produces output files containing the evaluation results. Using these output files, the results can be instantly viewed in the web app.

4 Features

4.1 Error type classification

Elevant automatically classifies each false positive and false negative into the following three categories and 15 subcategories, to provide detailed information about strengths and weaknesses of a linker.

NER false negatives are ground truth mentions which the linker did not link to an entity. They are divided into the following disjunct subcategories:

• Lowercased: The first letter in the mention is lower-cased. Linkers that rely on the upper case too much have many errors in this subcategory on benchmarks that contain lower-cased mentions.

• Partially included: Not lowercased and a subspan of the mention is linked to an entity. Often a less specific mention is recognized instead of a more specific one, e.g. recognizing "2022 World Cup" instead of "2022 World Cup".

• Partial overlap: Neither lowercased nor partially included and a span overlapping with the false negative is linked to an entity, e.g. recognizing "The Americans" instead of "The Americans".

• Other: Remaining undetected mentions.

NER false positives are predicted mentions not labeled in the ground truth. They are further divided into the following disjunct subcategories:

• Lowercased: The predicted mention is lowercased (no named entity) and does not overlap with any ground truth mention. These are often mentions of abstract entities, which appear in the knowledge base, but are usually not labeled in entity linking benchmarks, for example, *love (Q316)*.

• Ground truth entity unknown: The ground truth of the predicted mention is *Unknown*, which means that the true entity is not part of the knowledge base, but an entity from the knowledge base was predicted. Linkers that fail to produce *NIL* predictions have many errors in this subcategory.

• Wrong span: The predicted mention overlaps with a ground truth mention that has the same entity, but the spans do not match exactly.

• Other: Remaining false detections.

Disambiguation errors are NER true positives that are linked to the wrong entity. They count as false positives and false negatives. They are further divided into the following disjunct subcategories:

• Demonym: The mention is a demonym (i.e., it is contained in a list of demonyms from Wikidata), such as "German". Confusions between a country, the people from that country or the language spoken in that country fall into this category.

• Metonymy: The mention is a location name (for example, Berlin), but the ground truth is not (for example, *government of Germany (Q159493)*), yet the linker wrongly predicted the location.

• Partial name: The mention is a part of the ground truth entity's name, e.g., the last name of a person.

• Rare: The linker predicted the most popular candidate entity (with candidate sets derived by entity names and Wikipedia hyperlink texts, and popularity measured by the number of Wiki sites about an entity) instead of a less popular one.

• Other: Remaining disambiguation errors.

For linkers where we have access to the candidate sets, the following disambiguation error subcategories are reported. They overlap with the previous five subcategories.

• Wrong candidates: The ground truth entity is not contained in the candidate set.

• Multiple candidates: The ground truth entity is contained in the candidate set, but the wrong candidate was predicted.

4.2 Evaluation per entity type

Elevant assigns a type to each entity and computes precision, recall and F1 score per entity type. Many entity linking benchmarks contain more than the classic person, location and organization entities. We therefore chose 29 entity types that cover the entities in the included benchmarks well, yet are not too abstract to include many Wikidata entries that are not linked in the benchmarks. Example types are person, location, organization, languoid, taxon, brand, award, event and chemical entity (for the full list see Elevant's documentation on GitHub). The types are not restricted to *named* entities, but include other types of interest, such as profession, sport and color. A type t is assigned to an entity e, if t and e are connected in a manually corrected Wikidata dump via a property path that starts with an *instance of (P31)* relation and is followed by an arbitrary amount of *subclass of (P279)* relations. The types are configurable and detailed instructions for the configuration are given in Elevant's documentation.

4.3 Rich visualization

Elevant provides a rich and compact visualization of an entity linker's predictions in comparison to the ground truth labels; see Figure 2. Predictions are shown as highlighted text, while ground truth labels are shown as underlined text. Both predictions and ground truth labels are color-coded such that true positives, false positives, false negatives and unknown entities can be distinguished at a glance. On mouse-over, tooltips with additional information about the predicted entity or ground truth entity are shown, such as their Wikidata name and ID. When the user selects one of the error categories or entity types mentioned above, annotations that fall into the selected category are emphasized.

4.4 System comparison

Aside from letting the user compare the evaluation results of different entity linkers in various categories, Elevant comes with a feature to compare the predictions of two entity linkers for a selected benchmark side by side. This allows the user to closely and comfortably examine where differences in the evaluation results of two systems are coming from.

4.5 Automatic graph generation

For each category in the evaluation results table, Elevant can generate a graph that shows the results of all linkers over all benchmarks that are currently displayed in the table for that category. See Figure 3 for an example. Which linkers and benchmarks are included in the graph can be controlled by filtering the linkers and benchmarks that are to be included in the table as described in the next section.

4.6 Additional web app features

In addition to the prominent features described above, the Elevant web app comes with several features that improve overall usability. Each selectable component such as the experiment, error category or benchmark article has a corresponding URL parameter. The URL is automatically adjusted when a component is selected. This makes sharing the currently inspected results, e.g. the results of a particular linker for a particular error category on a particular benchmark as easy as copying and sharing the current URL.

When evaluating multiple entity linkers on multiple benchmarks, the evaluation results table can quickly become huge. In order to keep the focus on the currently most relevant results, Elevant has filter text fields which support regular expressions. Only linkers and benchmarks whose names match the filter texts are displayed in the table.

Our goal was to make the Elevant web app as intuitive as possible such that no additional resources would be necessary in order to understand and use it. To this end, the web app itself provides unobtrusive yet easily accessible explanations for its components. A mouseover button for example gives detailed explanations about the annotations such as the (already intuitive) color code. Hovering over the table header of an error category opens a tooltip that not only explains the corresponding error category but also gives an example for an entity linker error that falls into this category. Hovering over precision, recall or F1-score table cells opens a tooltip that shows the total numbers of true positives, false positives, false negatives and ground truth mentions for the corresponding category.

4.7 Included benchmarks

Elevant contains the following benchmarks:

- AIDA-CoNLL (Hoffart et al., 2011), a collection of 216 and 231 news articles from the 1990s for validation and testing.
- KORE50 (Hoffart et al., 2012), 50 difficult, hand-crafted sentences.
- MSNBC (Cucerzan, 2007), 20 news articles from 2007.
- MSNBC updated (Guo and Barbosa, 2018), a version of MSNBC without entities that do no longer exist in Wikipedia.
- DBPedia Spotlight (Mendes et al., 2011), 35 paragraphs from New York Times articles.

4.8 Included linkers

Elevant contains pre-computed results of the following entity linkers on the included benchmarks.

- TagMe (Ferragina and Scaiella, 2010)
- DBpedia Spotlight (Daiber et al., 2013)
- GENRE (Cao et al., 2021b)
- Efficient EL (Cao et al., 2021a)
- Neural EL (Gupta et al., 2017)
- Ambiverse (Seyler et al., 2018) (NER), (Hoffart et al., 2011) (NED)

TagMe and DBpedia Spotlight can be run out of the box with Elevant. For GENRE, Efficient EL and Neural EL, we provide code with an easy docker setup that yields results in a format supported by Elevant. Furthermore, Elevant can process any linking results file which is in NIF, the Ambiverse output format or a simple JSONL format as described in Section 3.2. These formats are also explained in detail in Elevant's documentation.

Additionally, we include a simple baseline that is based on prior probabilities computed from Wikipedia hyperlinks. The baseline uses the SpaCy (Honnibal et al., 2020) NER tagger with slight modifications (such as filtering out dates) to detect entity mentions. All entities with an alias that matches the mention text are considered as candidate entities for a mention. The aliases of an entity are the anchor texts of incoming intra-Wikipedia

hyperlinks to an entity's Wikipedia article, as well as the entity's Wikidata aliases. From these entity candidates, the entity that has most frequently been linked with the mention text in Wikipedia is predicted.

4.9 Extendability

New benchmarks or entity linking results can easily be added to Elevant if they are in one of the supported formats using Elevant's conversion scripts as described in Section 3.1 and Section 3.2. Additionally, support for other benchmark or entity linking result formats can be added with little effort. The process for implementing new format readers for benchmarks or entity linking results is explained in Elevant's documentation and existing format readers can be used as templates.

4.10 Easy knowledge base update

Elevant stores information about entities in several files that are generated from two sources: Wikidata and Wikipedia. The information extracted from Wikidata includes an entity's name, aliases, types and its corresponding Wikipedia URL. The information extracted from Wikipedia includes intra-Wikipedia link frequencies (how often is a hyperlink's anchor text in Wikipedia linked to a certain Wikipedia article) and Wikipedia redirects which are needed to reliably map Wikipedia entities to Wikidata. All of these files can either be downloaded from our servers or generated with three simple commands. The simplicity of the data generation allows for regular updates of the data.

4.11 Open source

Our code is open source (Apache License 2.0) and is available on GitHub [1]. A Docker setup allows an easy installation and usage. All links and a web demo are provided at `https://elevant.cs.uni-freiburg.de`.

5 Conclusion

Elevant is a powerful, general-purpose, easy-to-use system for the in-depth evaluation and comparison of a set of entity linkers on a given set of benchmarks. Typical evaluations of entity linking systems only provide aggregated figures like precision, recall and the F1 score. Elevant goes beyond this by providing a breakdown of the results by entity type and by error category, as well as an intuitive visualization of true positives, false negatives, and false positives on the concrete texts. This can help both practitioners (to understand for which kind of texts a given entity linker is suited) as well as researchers (to help understand in detail the particular weaknesses of their entity linker and try to improve those).

[1] `https://github.com/ad-freiburg/elevant/`

References

Adrian Brasoveanu, Giuseppe Rizzo, Philipp Kuntschik, Albert Weichselbraun, and Lyndon J. B. Nixon. 2018. Framing named entity linking error types. In *11th International Conference on Language Resources and Evaluation, LREC 2018*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021a. Highly parallel autoregressive entity linking with discriminative correction. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 7662–7669.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021b. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021*.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2007*, pages 708–716.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *9th International Conference on Semantic Systems, ISEM 2013*, pages 121–124.

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *19th ACM Conference on Information and Knowledge Management, CIKM 2010*, pages 1625–1628.

Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-dependent fine-grained entity type tagging. *CoRR*, abs/1412.1820.

Zhaochen Guo and Denilson Barbosa. 2018. Robust named entity disambiguation with random walks. *Semantic Web*, 9(4):459–479.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 2681–2690.

Benjamin Heinzerling and Michael Strube. 2015. Visual error analysis for entity linking. In *53rd Annual Meeting of the Association for Computational Linguistics and 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*, pages 37–42.

Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. 2012. KORE: keyphrase overlap relatedness for entity disambiguation. In *21st ACM International Conference on Information and Knowledge Management, CIKM 2012*, pages 545–554.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011*, pages 782–792.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Marcus Klang and Pierre Nugues. 2018. Linking, searching, and visualizing entities in wikipedia. In *11th International Conference on Language Resources and Evaluation, LREC 2018*.

Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *26th AAAI Conference on Artificial Intelligence, AAAI 2012*.

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. Dbpedia spotlight: shedding light on the web of documents. In *7th International Conference on Semantic Systems, I-SEMANTICS 2011*, pages 1–8.

Fabian Odoni, Philipp Kuntschik, Adrian M. P. Brasoveanu, and Albert Weichselbraun. 2018. On the importance of drill-down analysis for assessing gold standards and named entity linking performance. In *14th International Conference on Semantic Systems, SEMANTiCS 2018*, volume 137, pages 33–42.

Henry Rosales-Méndez, Aidan Hogan, and Barbara Poblete. 2019. Fine-grained evaluation for entity linking. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 718–727.

Dominic Seyler, Tatiana Dembelova, Luciano Del Corro, Johannes Hoffart, and Gerhard Weikum. 2018. A study of the importance of external knowledge in the named entity recognition task. In *56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, pages 241–246.

Michael Strobl, Amine Trabelsi, and Osmar R. Zaïane. 2020. WEXEA: wikipedia exhaustive entity annotation. In *12th Language Resources and Evaluation Conference, LREC 2020*, pages 1951–1958.

Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. 2015. GERBIL: general entity annotator benchmarking framework. In *24th International Conference on World Wide Web, WWW 2015*, pages 1133–1143.