RELD: A Knowledge Graph of Relation Extraction Datasets

Manzoor Ali[1] (✉), Muhammad Saleem[1], Diego Moussallem[1], Mohamed Ahmed Sherif[1], and Axel-Cyrille Ngonga Ngomo[1]

DICE group, Department of Computer Science, Paderborn University, Germany
manzoor@campus.uni-paderborn.de
saleem@informatik.uni-leipzig.de
diego.moussallem@uni-paderborn.de
mohamed.sherif,axel.ngonga@upb.de
https://www.dice-research.org/

Abstract. Relation extraction plays an important role in natural language processing. There is a wide range of available datasets that benchmark existing relation extraction approaches. However, most benchmarking datasets are provided in different formats containing specific annotation rules, thus making it difficult to conduct experiments on different types of relation extraction approaches. We present RELD, an RDF knowledge graph of eight open-licensed and publicly available relation extraction datasets. We modeled the benchmarking datasets into a single ontology that provides a unified format for data access, along with annotations required for training different types of relation extraction systems. Moreover, RELD abides by the Linked Data and FAIR principles. To the best of our knowledge, RELD is the largest RDF knowledge graph of entities relations from text, containing ∼1230 million triples describing 1034 relations, 2 million sentences, 3 million abstracts and 4013 documents. RELD contributes to a variety of uses in the natural language processing community, and distinctly provides unified and easy modeling of data for benchmarking relation extraction and named entity recognition models.

Keywords: Knowledge graph · Relation extraction · benchmarks · Natural language processing. · ontology · RDF

Resource Type: Datasets
Repository: https://github.com/dice-group/RELD
Homepage: https://manzoorali29.github.io/index.html
License: GNU General Public License v3.0
Endpoint: http://reld.cs.upb.de:8890/sparql
Dumps/Local endpoint: https://hobbitdata.informatik.uni-leipzig.de/RELD/
DOI: 10.5281/zenodo.7429677

1 Introduction

Relation extraction (RE) aims to predict a relation between named entities in a natural language text. For example, the sentence *"YouTube is an online video sharing and social media platform owned by Google."* suggests that the relation `owned_by` holds between the two named entities with labels `YouTube` and `Google`, respectively. RE plays an important role in many natural language processing (NLP) applications, including question answering [30], knowledge base creation and completion [24], information extraction, and event identification [15]. Owing to the importance of RE, various machine learning and rule-based approaches have been proposed to extract relations from natural language text [17,10]. Consequently, different RE datasets [21,22,5,7] are also available to benchmark existing RE approaches.

However, benchmarking RE systems with existing RE datasets leads to several challenges. First, the datasets are in *different formats*. For example, NYT-FB [21], and Wikipedia-Wikidata [22] are in JSON, WEBNLG [5] is in XML, and SemEval 2010 Task 8 [7] is in text form. Second, datasets contain different styles of annotations. For example, the relation `birthplace` has the representation `/people/person/place_of_birth` in the NYT-FB dataset, while in the WEBNLG dataset, the same `birthplace` relation is labeled with `birthPlace`. The different formats and representation require extra work to benchmark the RE systems across different datasets. Third, these datasets are primarily from a single source, which in turn might bias the results achieved by RE systems. For example, the NYT-FB dataset is extracted from New York Times articles, while Wikipedia is the source of FewRel and Wikipedia-Wikidata datasets. Fourth, some datasets have poor or missing annotations (relations, sentences, named entities). For example, NYT-FB has only 2.1 % of the training sentences annotated with corresponding Freebase triplets [25]. Fifth, some of these datasets do not provide a natural language representation of relations. For example, the `birthplace` relation only has the label `P19` in Wikipedia-Wikidata and FewRel [6] datasets. Sixth, some of these datasets focus on a limited number of relations and can hence only be used to benchmark very specific types of RE systems. For example, Google-RE has only four relations and targets binary relation extraction approaches. SemEval targets relation classification, and FewRel is for Few-shots [20] relation classification. To the best of our knowledge, no dataset is specialized for more than one type (binary, ternary, Few-shot, joint entity, relation extraction) of relation extraction. Finally, many of these datasets are imbalanced [23] and contain incorrect annotations [26]. All these shortcomings make it difficult to conduct a comprehensive evaluation of RE tools.

Keeping in view the aforementioned challenges, we present RELD, a single unified RDF representation of eight relation extraction and classification datasets. These datasets include well-known public and freely accessible[1] relation extraction datasets NYT-FB [21], Wikipedia-Wikidata [22], WEBNLG [5],

[1] We excluded datasets (e.g., TACRED) that are not freely available in this current version of the RELD. However, they can easily be included in the future.

SemEval 2010 Task 8 [7], Google-RE [16], FewRel [6], T-REx[4] and DocRed[29]. In RELD, each relation and corresponding sentence/document is modeled as a unique RDF resource, to which various statistics/annotations (for example, appearing entities, position of entities in a sentence) are attached in the form of properties. We used various NLP tools to attach the missing annotations. The resulting RELD RDF knowledge graph consists of 1,230 million triples, 1,034 unique relations, 2 million sentences, 3 million abstracts, and 4 thousand documents from different domains. To the best of our knowledge, RELD is the largest RDF dataset for relation extraction. We hope that the diversity of the relations, the unified model underlying the dataset and the improved relation annotations will contribute to easier and more comprehensive evaluations of RE systems.

The rest of this paper is structured as follows: Section 2 discusses the RELD data model. In Section 3, we outline a selection of use cases that illustrate the potential impact of our dataset and derive some requirements. Section 4 introduces the eight publicly available relation extraction datasets which are converted to RDF. We present details of the resulting RELD dataset and some statistics in section 5. In Section 6, we describe the availability and reusability of RELD. Section 7 provides some concrete examples of SPARQL queries over the RELD dataset based on the motivating use cases from Section 3, and Section 8 concludes.

2 RDF Data Model

Our goal is to create an RDF knowledge graph of existing relation extraction labeled datasets available in different formats. This section describes the RDF data model we utilize to capture the features (see Section 3) for underlying NLP tasks (relation extraction, named entity recognition, entity linking, etc.). The design of this data model was based on the following premises:

i. Generality: The data model must provide means to represent features of sentences, relations, and entities. The resulting dataset should allow use cases to be implemented based on the meta-data alone without needing to parse sentence text.
ii. Conciseness: Since datasets contain millions of sentences and entities, the data model should be concise to keep the overall dataset size manageable.
iii. Usability: SPARQL queries over the dataset should execute efficiently without requiring numerous joins or complex filters.
iv. Compatibility: IRIs should be made dereferenceable per Linked Data Principles. Furthermore, well-known vocabularies and ontologies should be reused where appropriate.

From these high-level requirements, we can derive a list of more concrete features that should be captured by the RELD data model:

Relation representation: Relations should be modelled in a uniform style that contains corresponding sentences along with their types. Furthermore,

equivalent relationships from different source datasets should be identified and interlinked.

Sentence features The dataset should describe the features used in an individual sentence (e.g., entities position, entities direction) in such a manner that a sentence can be filtered according to the features they use/omit. Similarly, the number of sentences using a single characteristic can be determined.

Sentence statistics The sentence metadata should likewise capture high-level information about the size and "complexity" of the sentences in terms of number of tokens, entity position, and number of entities tokens variables within a single sentence etc.

Named entities feature The available named entities in a sentence should be identified along with additional statistics (e.g., types, labels). The correctness of the identified named entities is also key.

In Figure 1, we provide an overview of the core of the schema for the RELD knowledge graph data model[2]. Listing 1.1 shows all the vocabularies used in RELD while listing 1.2 provides an example[3] output of the RELD knowledge graph.

Listing 1.1: List of all used vocabularies in RELD

```
@prefix reld: <http://reld.dice-research.org/schema/> .
@prefix reldr: <http://reld.dice-research.org/resource/> .
@prefix reldp: <http://reld.dice-research.org/property/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix freebase: <http://rdf.freebase.com/ns> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix ps: <http://www.wikidata.org/prop/statement/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix schema: <http://schema.org/> .
@prefix dicom: <http://purl.org/healthcarevocab/v1> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix void: <http://rdfs.org/ns/void#> .
@prefix bibtex: <http://purl.org/net/nknouf/ns/bibtex#> .
@prefix dcat: <http://www.w3.org/ns/dcat> .
@prefix prof: <http://www.w3.org/ns/dx/prof/hasToken> .
```

Dataset: As a practical design decision, we create dataset instances for each dataset, whereby a dataset instance represents a single dataset that we consider for conversion to RDF for RELD. The `reld:Dataset` class contains the basic information about the datasets such as the homepage URL, the task for which the dataset is known, the type of the dataset such as document type or sentence type,

[2] The detail information of schema, i.e., object properties, data properties, classes are available on RELD homepage.

[3] Due to page size limitation, some details and extra instances are truncated from Listing 1.2

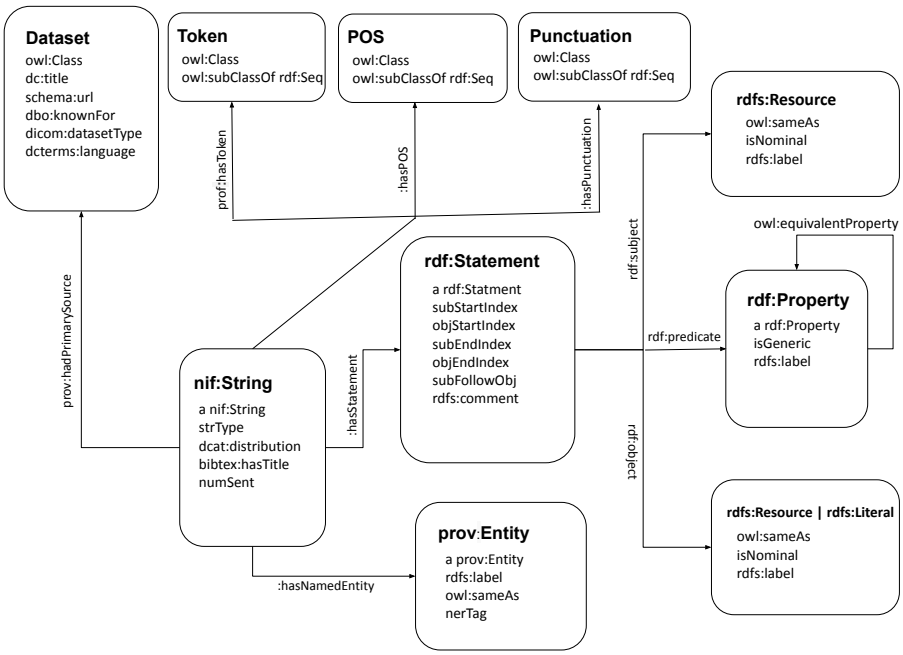


Fig. 1: RELD Data Model

title, and language of the dataset[4]. Every instance of `nif:String` (discussed next) linked with the dataset as a `prov:hadPrimarySource`.

String: For reusability, we use `nif:String` to model each sentence/document of the source dataset. The `nif:String` class avoids the ambiguity between sentences and documents. String class has a property `:strType` that differentiates a string as a sentence or document. Every instance of `nif:String` has an IRI <http://reld.dice-research.org/resource/S-147375> that contains a unique ID. Some datasets, e.g., DocRed contain titles for paragraphs or documents that we map in the RELD model using `bibtex:hasTitle`. RELD uses `dcat:distribution` to know the original distribution (train, test, validation) of a string in the parent datasets. Additionally, the `:numSent` property shows the number of sentences in a paragraph or document if mentioned in the source dataset.

Token, POS and Punctuation: An instance of a string connects to a `:Token` and `:POS` classes using the `prof:hasToken` and `:hasPOS` properties, respectively. `Token` and `POS` are the subclasses of `rdf:Seq`. An instance of a `Token` contains the words of a sentence or document in a sequence, where every token represents a word or punctuation mark in the same order as it appears in the original text.

[4] We use void vocabulary to describe different metadata of the dataset

Listing 1.2: An example listing of RELD knwoledge graph

```
# String Instance
reldr:S-147375 a nif:String ;
    reldr:hasPOS reldr:posSeq147375 ;
    reldr:hasPunctuation reldr:puncSeq147375 ;
    reld:hasNamedEntity reldr:1, reldr:2014, reldr:50000, reldr:koln ;
    reld:hasStatement reldr:Stmt1473750, reldr:Stmt1473751 ;
    reld:strType "sentence"^^xsd:string ;
    dcat:distribution "train"^^xsd:string ;
    prof:hasToken reldr:token_147375 ;
    prov:hadPrimarySource reldr:ds_05 .
# Dataset Instance
reldr:ds_05 a reld:Dataset ;
    dbo:knownFor "natural_language_generation"^^xsd:string ;
    dc:title "WebNLG"^^xsd:string ;
    dcterms:language "en"^^xsd:string ;
    dicom:datasetType "sentence"^^xsd:string ;
    schema:url <https://webnlg-challenge.loria.fr/> .
# Statement Instance
reldr:Stmt1473751 a rdf:Statement ;
    reld:objEndIndex 11 ;
    reld:objStartIndex 11 ;
    reld:subEndIndex 2 ;
    reld:subFollowObj false ;
    reld:subStartIndex 2 ;
    rdf:object reldr:50000 ;
    rdf:predicate reldr:numberOfMembers ;
    rdf:subject reldr:1_fc_k_ln .
# Token Instance
reldr:token_147375 a reld:Token ;
    rdf:_0 "2014"^^xsd:token ;
    rdf:_1 "saw"^^xsd:token ;
    rdf:_10 "have"^^xsd:token ;
    rdf:_11 "50000"^^xsd:token ;
    rdf:_12 "members"^^xsd:token ;
    rdf:_13 "."^^xsd:token ;
    rdf:_2 "1"^^xsd:token ;
    rdf:_3 "."^^xsd:token ;
    rdf:_4 "FC"^^xsd:token ;
    rdf:_5 "Koln"^^xsd:token ;
    rdf:_6 "participating"^^xsd:token ;
    rdf:_7 ","^^xsd:token ;
    rdf:_8 "and"^^xsd:token ;
    rdf:_9 "they"^^xsd:token .
# POS Instance
reldr:posSeq147375 a reld:POS ;
    rdf:_0 "CD"^^xsd:string ;
    rdf:_1 "VBD"^^xsd:string ;
    ...
    rdf:_12 "NNS"^^xsd:string ;
    rdf:_13 "."^^xsd:string ;
# Punctuation Instance
reldr:puncSeq147375 a reld:Punctuation ;
    rdf:_0 "."^^xsd:string ;
    rdf:_1 ","^^xsd:string ;
    rdf:_2 "."^^xsd:string .
# Object Instance
reldr:50000 a rdfs:Literal, prov:Entity ;
    rdfs:label "50000"^^xsd:string ;
    reld:nerTag "CARDINAL"^^xsd:string .
# Entity Instance
reldr:FC_koln a prov:Entity ;
    rdfs:label "koln"^^xsd:string ;
    reld:nerTag "ORG"^^xsd:string ;
    owl:sameAs dbr:FC_koln .
# Property Instance
reldr:numberOfMembers a rdf:Property ;
    rdfs:label "numberOfMembers"^^xsd:string .
# Subject Instance
reldr:1_fc_k_ln a rdfs:Resource ;
    rdfs:label "1_FC_K_ln"^^xsd:string .
```

Similarly, a `POS` instance represents a part of the speech tag for each corresponding token in the sentence/document. Listing 1.2 shows an example of a `:Token` and `:POS`. Likewise, in `:Token` and `:POS` classes, the `:Punctuation` class maps all the punctuation[5] of the original text. It is also the subclass of the `rdf:Seq` class. `:hasPunctuation` property links a `nif:String` to `:Punctuation`.

Statement: An instance of a `nif:String` may contain one or more annotated statements linked with them by a property `:hasStatement`. A Statement consists of a subject of type `rdfs:Resource`, an object of type `rdfs:Resource` or `rdfs:Literal`, and a predicate as `rdf:Property`. Properties like `:subStartIndex`, `:subEndIndex`, `:objStartIndex`, and `:objEndIndex` show the position of subject and object entities (also called head entity and tail entity) in a sentence or document. We annotate a statement with `:subFollowObj` boolean property as True if an object entity appears before a subject entity in the text, False otherwise. Remaining properties shown in the model diagram map further metadata related to each statement in the source dataset.

A subject and object both have a property `:isNominal` which indicates the sort of entity involved in a relation. For example, the sentence *"The suspect dumped the dead < /e2 >body< /e2 > into a local </e2>reservoir </e2>. "* and the relation *"Entity-Destination(*$e1, e2$*)"* in SemEval2010 dataset has nominal entities. We take an open-world assumption and keep it True if we know that an entity is nominal.

To consolidate the subject and the object, we did not use any ID for them that enables multiple sentences pointing to a single subject or an object. In addition, the subject entity can appear as an object entity in another text and vice versa if both have type `rdfs:Resource`. To deal with the lexical variability (same entity but different representations), RELD keeps all of them separately but uses the `owl:sameAs` property to link the same entity to similar entities in other knowledge bases. For example, *Obama* and *Barack Obama*, we keep them separately, but both entities have `owl:sameAs` property `dbr:Barack_Obama`.

`rdf:Property:` of a statement maintains the annotation of the relation in the source dataset. We disambiguate relations within each dataset, and if two relations represent the same relation, we link them using the `owl:equivalentProperty`. For example, the WEBNLG dataset has two different annotations for affiliation property *affiliation* and *affiliations*. To preserve the original annotation, RELD keeps both representations and links them using the `owl:equivalentProperty`. We have manually aligned similar properties based on the similarity information from the literature [28,13]. In the next version, we plan to use LIMES[14] and MAG[12] to score the similarity among properties and improve the linking. For relations like "Entity-Destination(e1,e2)" discussed earlier, RELD introduces the `:isGeneric` property. In the case of `:isGeneric` property, we also take an open-world assumption.

`Entity:` The number of entities plays an essential role in the relation prediction in a natural language text, whether it is not directly involved in the actual

[5] We use NLTK [9] for tokenization, parts of speech tagging, and punctuation

relation [26]. Relation extraction datasets do not provide this information. To overcome this issue and increase the use cases of RELD, we annotate the text for named entities using Spacy [8]. A `String` instance can have zero or more named entities that may or may not be involved directly in a relation. RELD maps this information using the `prov:Entity` class. Using this information, the user can generate a custom benchmark that includes the required numbers of named entities[2]. Furthermore, the `owl:sameAs` property links the entity with other linked datasets.

To stick with conciseness, we avoid annotating features that affect the overall score of a relation extraction approach, but a SPARQL query can derive it from other basic annotations. For example, using the properties like `:subStartIndex`, `:subEndIndex`, `:objStartIndex`, and `:objEndIndex` in SPARQL user can retrieve the features such as the number of tokens before the subject entity, after the object entity, or between the two entities[1,3].

3 Impact

In this section, we cover several use cases for RELD to explain the potential impact and usage of the knowledge graph. These are the use cases we foresee going forward:

UC1 Custom Benchmarks The RELD dataset can be used to generate customized, use-case-specific benchmarks (called micro-benchmarks) by selecting the desired number of relations, length, and size of sentences with the desired number of mentioned entities within a sentence. Recently, the RELD dataset has already been used to generate micro-benchmarks according to the user-specified criteria [2]. We provided a sample query in Section 7 that shows the use-case-specific benchmarking of the RELD dataset.

UC2 Balanced Dataset Selection For better performance of a model, it requires a balanced dataset to train, where each relation has a similar number of sentences [23]. Using RELD, a balanced sub-dataset generation requires the execution of a single SPARQL query with desired filters. The sub-dataset can train a machine-learning algorithm on a large scale. Section 7 has a sample query that generates a balanced sub-dataset. In addition, the RELD dataset can be used for few-shot relation extraction [19], where a given relation is only found in a few sentences.

UC3 Generic model RELD contains relations and sentences of various types from diverse domains; hence, on top of RELD, we can train and test generic RE models.

UC4 Other NLP tasks In addition, the schema and data of RELD enable it for other underlying natural language processing tasks, such as:

- **Causal relation classification:** Properties, i.e. `:isGeneric`, `:isNominal` enable RELD to be used for classifying casual relations. RELD contains the SemEval2010 Task 8, a causal relation classification dataset as named graph. Furthermore, the RELD schema can easily incorporate other such datasets.

- **Natural language generation:** The representation of entities and relations in statements makes the RELD schema compatible with the natural language generation datasets. Also, it contains the WebNLG dataset to fulfill this task.
- **Named entity recognition/disambiguation:** Entities annotation and linking with other knowledge bases add NER and NED use-cases to the RELD domain. Also, researchers can exploit RELD knowledge graphs for joint entity and relation extraction models.
- **Document-based Relation extraction:** Apart from sentence-based RE, RELD concisely includes document-based RE datasets, which can train an RE model on documents instead of sentences.

These are only a few use cases, and we can firmly imagine others.

4 Current Used Datasets

In this section, we briefly discuss the datasets that we used for building the RELD knowledge graph. In the current version, we only included those datasets that are publicly available, free of charge, and their license permits us to reuse the data in a different representation. To this end, we excluded datasets that are not free of charge. However, we are planning to include paid datasets (e.g., TACRED [31], ACE2005 [27]) in the future if their license permits. Currently, we also ignore datasets that target specific-purpose relation extraction, such as ChemProt [18], which is for biomedical relation extraction.

We wrote scripts to extract and normalize data from each dataset and map to the target schema explained in section 2. In the current state, RELD consists of eight state-of-the-art open-sourced relation extraction datasets:

Wikipedia-Wikidata (WW) [22] WW dataset is extracted from Wikipedia text and aligned with the Wikidata relations. It is the second-largest dataset in RELD and consists of train, test, and validation sets in JSON format. The primary task of this dataset is the multi-relation (a single sentence can contain multiple relationships) extraction. In RELD, we keep the original annotation of the WW that are Wikidata identifiers. We also exploit Wikidata for the natural language representation of each relation and map it to the `rdfs:label` property.

FewRel [6] Like WW, FewRel's primary source is also Wikipedia for text and Wikidata for annotation. The primary task of this dataset is a few-shot relation classification. It is the only dataset in RELD that contains a balanced number of sentences (i.e., 700) for each relation. Due to the same sources, the basic structure of this dataset is also similar to WW.

NYT-FB [21] A dataset primarily created for distant supervision-based relation extraction is one of the most commonly used datasets in the relation extraction community. The dataset is extracted from New York Times articles and aligned with the freebase dataset. This dataset contains 24 relations, of which 50% are

also available in other datasets, while the remaining 50% are unique.

WEBNLG [5] The primary purpose of this dataset is natural language generation. The dataset contains 354 relations that include 'Other' (a sentence may have a relationship, but that is not part of the defined set) relation. This dataset comprises the automatically generated sentences from DBpedia triples, where a sentence contains a range of 1 to 7 triples. This multi-triple nature of sentences in the WEBNLG dataset makes it perfect for the multi-relation extraction task.

Google-RE [16] Google relation extraction dataset consists of four relations represented in JSON format. The primary task of this dataset is binary relation extraction from sentences. Similar to the NYT-FB dataset, this dataset is also aligned with freebase. This dataset does not explicitly specify the train, test, and validation sets. Instead of only sentences, this dataset contains paragraphs for a single relation, so a relationship may appear between two entities that are not necessarily in a single sentence. The average length of the number of tokens is relatively higher than other datasets, which makes Google-RE a challenging dataset.

SemEval 2010 Task 8 [7] Instead of relation extraction, relation classification is the primary task of the SemEval2010 Task 8 dataset. It differs from the other relation extraction datasets as it does not contain a relation between two named entities. But it consists of sentences that have a generic relationship between two nominals. The sentence structure decides the subject and object entities, and the relationship depends on the direction of the two entities. We put `:subFollowsObj` in the RELD model to identify the order of entities in a sentence and identify the subject and object entities' position in the sentence. Furthermore, RELD handles generic relations using the `:isGenric` property and nominal entities using the `:isNominal` property.

DocRed [29] Unlike other datasets, DocRed is used for relation extraction from documents instead of sentences. This dataset is also different from the other relation extraction datasets because it consists of paragraphs (called documents) instead of sentences. It may have one or more relations between two named entities and also has a title. RELD has a property `:title` to identify the same document of various sentences and has a property `:numSent` which shows the number of sentences in a document.

T-REx [4] T-REx is the largest dataset mapped to RELD so far. It consists of more than six million sentences and 685 relations. Like FewRel and WW, its primary source is Wikipedia abstracts and Wikidata entities. We represent T-REx as a document-based dataset because of its similarity with the DocRed dataset.

5 RELD Dataset Statistics

Table 1 shows the relation extraction type, origin, number of relationships, and the total number of sentences/docs/abstracts in each selected dataset. It is worth noting that each dataset targets a particular kind of relation extraction that limits the use of a single dataset only for a single type of relation extraction. The RELD dataset contains a good variety in terms of the number of sentences corresponding to different relations. On average, WW provides the highest (5030 sentences) number of sentences per relation, followed by NYT-FB (4638), Google (4237), SemEval (1071), FewRel (700), WEBNLG (218), respectively. Table 2 shows the distribution of the relations and the corresponding sentences according to the train, test, and validation sets.

Table 1: Basic information of all the datasets used in RELD.

Datasets	RE Type	Source	# relation	# Sentences
WEBNLG	NL generation	DBpedia	354	53,786
NYT-FB	Sentence	Web-Freebase	24	111,327
FewRel	Few-shots	Wikipedia-Wikidata	80	56,000
SemEval	Classification	Crowd sourced	10	10,717
Google	Sentence	Web	4	16,948
WW	Sentence	Wikipedia-Wikidata	352	1,770,721
DocRed	Document	Wikipedia-Wikidata	96	4013 docs
T-REx	Sentence/Document	Wikipedia-Wikidata	685	3M abstracts

Table 2: Distribution of relations and sentences/documents in train, test and validation in different dataset. D represents documents, while M for a million

	Train		**Validation**		**Test**	
Dataset	relations	sent/docs	relations	sent/docs	relations	sent/docs
WEBNLG	246	74,779	186	72,719	246	80,710
NYT-FB	24	111,327	22	111,324	22	111,324
FewRel	64	44,800	16	11,200	0	0
SemEval	10	8,000	0	0	10	2,717
Google	4	16,948	0	0	0	0
WW	352	1,770,721	352	1,770,721	352	1,770,721
DocRed	96	3,053D	96	1000D	96	1000D
T-REx	685	6.02M/3M	0	0	0	0
Overall	**1,481**	**8.04M/3M˜**	**672**	**0.36M/1K**	**726**	**1.96M/1K**

Table 3: Basic RDF statistics of the RELD datasets. SUB & OBJ represents subject and objects respectively, while R for resource and L for literal

Dataset	Triples	Resources	Named Entities	sameAs	Statements
SemEval	0.68M	7,592	8,941	907	10,717
NYT-FB	7.85M	14,663	365,373	17,179	111,327
FewRel	4.75M	71,940	231,122	23,969	56,000
WebNLG	1.02M	2,555	42,473	827	30,849
Google-RE	3.13M	20,028	169,319	14,169	14,458
WW	78.16M	515,422	3,701,186	512,010	1,770,721
DocRed	2.55M	25,675	84,066	9,408	50,503
T-REx	1132.08M	-	43,897,838	4,416,214	20,834,823

Table 3 shows various RDF-related features for each source dataset. In total, we have 1230 million triples, 48.5 million named entities, 5 million `owl:sameAs` links, and 23 million statements included in the final RELD dataset. Finally, Table 4 shows information about the structure and complexity of the sentences of the selected datasets, where average Before refers to the average number of tokens in the sentence before the subject entity of a relation. Similarly, AVG Between refers to the number of tokens between the subject and object entity, and AVG after refers to the number of tokens after the object entity. Clearly, the Google-RE dataset is more complex in terms of the number of tokens per sentence. Despite complex sentence structures, RE systems perform better (in terms of F scores) on Google-RE dataset [11]. This indicates that evaluation based on a single dataset with a small number of relations (4 in google dataset) might not sufficiently stress the RE systems. In total, there are 125 overlapping relations in the selected datasets. Overlapping relations have different representations in each dataset. For example, The Wikipedia-Wikidata dataset has a relation P19 that represents the place of birth; the same relationship is presented as /people/person/place_of_birth in NYT-FB and Google-RE, and is called birthPlace in the WEBNLG dataset. We use the `:equivalentProperty` to highlight the same relations from different datasets. Table 5 shows the top five relationships which appear in more than one dataset. Figure 2 shows the number of overlapping relationships among the three sentence-based datasets.

The `:equivalentProperty` increases the number of sentences for a given relation because it makes RELD capable of discovering all sentences that contain the given relationship in a different form. Figure 3 shows the range of sentences for a different number of relations that are available in more than one dataset. For example, Figure 3a shows that 385 relationships have less than 100 sentences. However, in Figure 3b this number reduces to 167 relationships by using `:equivalentProperty` information, which also increases the number of sentences for those relations. Furthermore, some datasets contain similar relations with different names, like *affiliation* vs. *affiliations* or *Leader* vs. *Leader-*

Name in the WEBNLG dataset. We identify 22 such relations manually and use `:equivalentProperty` to relate all those relations.

Table 4: Tokens-related information from all the sentence based datasets.

Datasets	Avg Tokens	Tokens > 30	Avg Before	Avg Between	Avg After
WEBNLG	27	37%	2.6	7	14
NYT-FB	39	70%	12.6	9	14
FewRel	24	23%	6.5	6	7
SemEval	19	10%	5.1	4	7
Google	74	79%	0.07	4	68
WW	24	16%	7.3	6	7

Table 5: Top 5 relations that occurred in more than one dataset in RELD

Relation	WEBNLG	NYT-FB	FewRel	SemEval	Google	WW	DocRed	T-REx
birthDate	✓	✗	✗	✗	✓	✓	✓	✓
birthPlace	✓	✓	✗	✗	✓	✓	✓	✓
deathPlace	✓	✓	✗	✗	✓	✓	✓	✓
nationality	✓	✓	✓	✗	✗	✓	✓	✓
country	✓	✓	✓	✗	✗	✓	✓	✓
location	✓	✓	✓	✗	✗	✓	✓	✓

6 Resource Availability, Reusability, Sustainability

The resource is publicly available from the homepage, which contains the complete source code, data, and documentation. The homepage also links to the corresponding RELD ontology. The same home page will be used for sustainability and adding future datasets into the RELD. Paderborn Center for Parallel Computing PC2 will sustain the RELD resources. PC2 provides computing resources and consultation regarding their usage; to research projects at Paderborn University and external research groups. The Information and Media Technologies Center (IMT) at Paderborn University also provides a permanent IT infrastructure to host the RELD project. The open-source code available on GitHub is easily extendable to convert other datasets in the future. The RELD dataset is publicly available from the SPARQL endpoint, where the user can execute a SPARQL query for desired output.

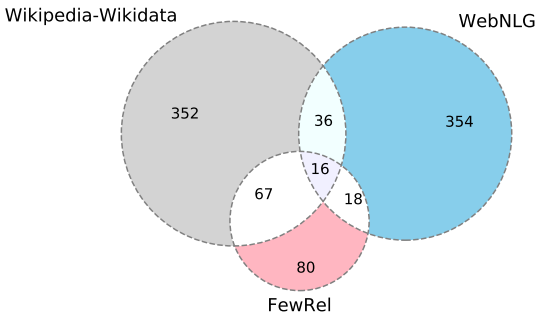
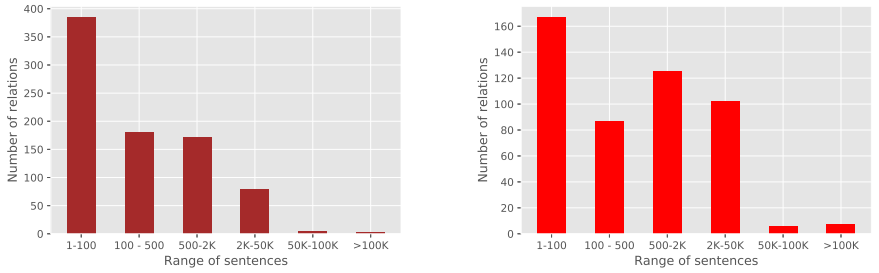


Fig. 2: Venn diagram for the number of overlapping relations in the three sentence-based datasets



(a) Without `owl:equivalentProperty` (b) With `owl:equivalentProperty`

Fig. 3: Number of relations in range of sentences in RELD

7 RELD in Practice

We have made the RELD dataset available through three media: (i) dereferenceable Linked Data, (ii) flat dumps, and (iii) a SPARQL endpoint. In this section, we provide a few concrete queries that can be issued against the RELD SPARQL endpoint to derive insights relevant to some use cases discussed in Section 3.

UC1 Facilitating Custom Benchmark Generation: RELD can help users to generate custom benchmark meeting defined criteria for a given use case. Listing 1.3 is an example SPARQL query over RELD that selects a benchmark, each containing less than 50 tokens and more than 10 entities and more than four relations. RELD-based microbenchmarking framework for RE systems is presented in [2], where users can generate customized and more representative benchmarks by using different clustering techniques.

Listing 1.3: **UC1:** Generate benchmark of having sentences length less than 50, and other required features

```
PREFIX reld: <http://reld.dice-research.org/schema/>
PREFIX nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#>
PREFIX prof: <http://www.w3.org/ns/dx/prof/>

SELECT DISTINCT
?sent (count(?t) as Tokens) (count(?e) as ?Entities) (count(?stmt) as ?
    Statment)
WHERE
          {?sent a nif:String;
                reld:hasStatement ?stmt;
                reld:hasNamedEntity ?e;
                prof:hasToken ?token.
            ?token ?p ?t.
}
GROUP BY ?sent
HAVING(COUNT(?stmt) > 4 && COUNT(?e) > 10 && COUNT(?t) < 50)
```

UC2 Balanced Dataset To generate a balanced dataset, where all the selected relations should have the same number of relevant sentences. Listing 1.4 selects a benchmark of relations where each relationship has exactly 700 annotated sentences that contain this relation.

Listing 1.4: **UC2:** A balance dataset of relations each having 700 sentneces that contain the given relation.

```
PREFIX reld: <http://reld.dice-research.org/schema/>
PREFIX nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#>
PREFIX prof: <http://www.w3.org/ns/dx/prof/>

SELECT DISTINCT ?properties  COUNT(?sent)
 WHERE {
        ?sent a nif:String;
        reld:hasStatement ?stmt.
        ?stmt rdf:predicate ?properties.
        }
GROUP BY ?properties
HAVING( COUNT(?sent) = 700)
```

8 Conclusion and Future Work

We presented RELD, to the best of our knowledge, the first publicly available knowledge graph for relation extraction that describes sentences with their annotation and labeled relations. We discussed various use cases for RELD with a detailed description of the model and basic statistics about the used datasets. Furthermore, we hope RELD can facilitate the benchmarking of the relation extraction tools. We are targeting to incorporate multilingual datasets to increase their use cases. The initial processing of multilingual datasets has already been in the final stages, and we will announce the integration to RELD on the project homepage. In addition, paid datasets such as TACRED [31] are also under consideration in the future. Finally, we plan to train a generic relation extraction model which extends the scope in terms of number relations and variability.

References

1. Agichtein, E., Gravano, L.: Snowball: Extracting relations from large plain-text collections. In: Proceedings of the fifth ACM conference on Digital libraries. pp. 85–94 (2000)
2. Ali, M., Saleem, M., Ngomo, A.C.N.: Rebench: Microbenchmarking framework for relation extraction systems. In: The Semantic Web – ISWC 2022: 21st International Semantic Web Conference, Virtual Event, October 23–27, 2022, Proceedings. p. 643–659. Springer-Verlag, Berlin, Heidelberg (2022), https://doi.org/10.1007/978-3-031-19433-7_37
3. Batista, D.S., Martins, B., Silva, M.J.: Semi-supervised bootstrapping of relationship extractors with distributional semantics. In: In Empirical Methods in Natural Language Processing. ACL (2015)
4. Elsahar, H., Vougiouklis, P., Remaci, A., Gravier, C., Hare, J., Laforest, F., Simperl, E.: T-rex: A large scale alignment of natural language with knowledge base triples. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (2018)
5. Gardent, C., Shimorina, A., Narayan, S., Perez-Beltrachini, L.: Creating training corpora for NLG micro-planners. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 179–188. Association for Computational Linguistics, Vancouver, Canada (Jul 2017). https://doi.org/10.18653/v1/P17-1017, https://aclanthology.org/P17-1017
6. Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z., Sun, M.: Fewrel:a large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In: EMNLP (2018)
7. Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., Szpakowicz, S.: SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In: Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 33–38. Association for Computational Linguistics, Uppsala, Sweden (Jul 2010), https://aclanthology.org/S10-1006
8. Honnibal, M., Montani, I.: spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. To appear **7**(1), 411–420 (2017)
9. Loper, E., Bird, S.: Nltk: the natural language toolkit. arXiv preprint cs/0205028 (2002)
10. Martinez-Rodriguez, J.L., Hogan, A., Lopez-Arevalo, I.: Information extraction meets the semantic web: a survey. Semantic Web **11**(2), 255–335 (2020)
11. Moreira, J., Oliveira, C., Macêdo, D., Zanchettin, C., Barbosa, L.: Distantly-supervised neural relation extraction with side information using bert. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–7 (2020). https://doi.org/10.1109/IJCNN48605.2020.9206648
12. Moussallem, D., Usbeck, R., Röeder, M., Ngomo, A.C.N.: Mag: A multilingual, knowledge-base agnostic and deterministic entity linking approach. In: Proceedings of the Knowledge Capture Conference. pp. 1–8 (2017)
13. Nadgeri, A., Bastos, A., Singh, K., Mulang', I.O., Hoffart, J., Shekarpour, S., Saraswat, V.: KGPool: Dynamic knowledge graph context selection for relation extraction. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 535–548. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.findings-acl.48, https://aclanthology.org/2021.findings-acl.48

14. Ngonga Ngomo, A.C., Sherif, M.A., Georgala, K., Hassan, M., Dreßler, K., Lyko, K., Obraczka, D., Soru, T.: LIMES - A Framework for Link Discovery on the Semantic Web. KI-Künstliche Intelligenz, German Journal of Artificial Intelligence - Organ des Fachbereichs "Künstliche Intelligenz" der Gesellschaft für Informatik e.V. (2021), https://papers.dice-research.org/2021/KI_LIMES/public.pdf
15. Ning, Q., Feng, Z., Roth, D.: A structured learning approach to temporal relation extraction. arXiv preprint arXiv:1906.04943 (2019)
16. Orr, D.: 50,000 lessons on how to read: a relation extraction corpus. Online: Google Research Blog **11** (2013)
17. Pawar, S., Palshikar, G.K., Bhattacharyya, P.: Relation extraction: A survey. arXiv preprint arXiv:1712.05191 (2017)
18. Peng, Y., Yan, S., Lu, Z.: Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets. arXiv preprint arXiv:1906.05474 (2019)
19. Qu, M., Gao, T., Xhonneux, L.P., Tang, J.: Few-shot relation extraction via Bayesian meta-learning on relation graphs. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 7867–7876. PMLR (13–18 Jul 2020), https://proceedings.mlr.press/v119/qu20a.html
20. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning (2016)
21. Riedel, S., Yao, L., McCallum, A., Marlin, B.M.: Relation extraction with matrix factorization and universal schemas. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 74–84 (2013)
22. Sorokin, D., Gurevych, I.: Context-aware representations for knowledge base relation extraction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 1784–1789. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017). https://doi.org/10.18653/v1/D17-1188, https://aclanthology.org/D17-1188
23. Sui, D., Chen, Y., Liu, K., Zhao, J., Zeng, X., Liu, S.: Joint entity and relation extraction with set prediction networks. arXiv preprint arXiv:2011.01675 (2020)
24. Surdeanu, M., Tibshirani, J., Nallapati, R., Manning, C.D.: Multi-instance multi-label learning for relation extraction. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. pp. 455–465 (2012)
25. Tran, T.T., Le, P., Ananiadou, S.: Revisiting unsupervised relation extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7498–7505. Association for Computational Linguistics, Online (Jul 2020), https://www.aclweb.org/anthology/2020.acl-main.669
26. Tran, T.T., Le, P., Ananiadou, S.: Revisiting unsupervised relation extraction (2020). https://doi.org/10.48550/ARXIV.2005.00087, https://arxiv.org/abs/2005.00087
27. Walker, C., Strassel, S., Medero, J., Maeda, K.: Ace 2005 multilingual training corpus. Linguistic Data Consortium, Philadelphia **57**, 45 (2006)
28. Wang, Y., Yu, B., Zhang, Y., Liu, T., Zhu, H., Sun, L.: TPLinker: Single-stage joint extraction of entities and relations through token pair linking. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 1572–1582. International Committee on Computational Linguistics, Barcelona, Spain (Online) (Dec 2020). https://doi.org/10.18653/v1/2020.coling-main.138, https://aclanthology.org/2020.coling-main.138

29. Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., Liu, Z., Huang, L., Zhou, J., Sun, M.: Docred: A large-scale document-level relation extraction dataset. arXiv preprint arXiv:1906.06127 (2019)
30. Yu, M., Yin, W., Hasan, K.S., Santos, C.d., Xiang, B., Zhou, B.: Improved neural relation detection for knowledge base question answering. arXiv preprint arXiv:1704.06194 (2017)
31. Zhang, Y., Zhong, V., Chen, D., Angeli, G., Manning, C.D.: Position-aware attention and supervised data improve slot filling. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 35–45 (2017)