REBench: Microbenchmarking Framework for Relation Extraction Systems

Manzoor Ali[1] (✉), Muhammad Saleem[2], and Axel-Cyrille Ngonga Ngomo[1]

[1] DICE group, Department of Computer Science, Paderborn University, Germany
manzoor@campus.uni-paderborn.de
axel.ngonga@upb.de
https://www.dice-research.org/

[2] AKSW Research Group, University of Leipzig, Germany
saleem@informatik.uni-leipzig.de

Abstract. In recent years, several relation extractions (RE) models have been developed to extract knowledge from natural language texts. Accordingly, several benchmark datasets have been proposed to evaluate these models. These RE datasets consisted of natural language sentences with a fixed number of relations from a particular domain. Albeit useful for general-purpose RE benchmarking, they do not allow the generation of customized microbenchmarks according to user-specified criteria for a specific use case. Microbenchmarks are key to testing the individual functionalities of a system and hence pinpoint component-based insights. This article proposes REBench, a framework for microbenchmarking RE systems, which can select customized relation samples from existing RE datasets from diverse domains. The framework is flexible enough to choose relation samples of different sizes and according to the user-defined criteria on essential features to be considered for RE benchmarking. We used various clustering algorithms to generate microbenchmarks. We evaluated the state-of-the-art RE systems using different RE benchmarking samples. The evaluation results show that specialized microbenchmarking is crucial for identifying the limitations of various RE models and their components.

Resource Type: Evaluation benchmarks or Methods
Repository: https://github.com/dice-group/REBench
License: GNU General Public License v3.0

Keywords: Microbenchmark · Relation Extraction · Clustering Algorithm.

1 Introduction

Relation extraction (RE) systems extract the relationship between two named entities from natural language texts. The named entities are often pre-annotated, and the task is to determine the relationship between the entities. There have a wide range of applications of RE including knowledge base creation [25], event generation [14], and question-answering approaches [33]. In recent years, several novel approaches have been proposed to extract relations, including rule-based [22] and machine learning [27,11,26] approaches. These approaches operate in different environments, such as supervised, semi-supervised, distant-supervised, and unsupervised [17].

Research Gap: Several datasets such as NYT-FB [18], TACRED [36], WEB-NLG [8], Wikidata RE [21], and SemEval-2010 [10] have been proposed to benchmark RE systems. These datasets (Table 1) contain a fixed number of relations from a particular domain and are sufficient to test the overall performance of the RE system in terms of precision and recall. However, they do not allow generation of use-case-specific benchmarking based on user specified-criteria. For example, a user may be interested in testing a given RE system using a benchmark containing only `binary relations` with `a fixed number of sentences` and `more than three named entities in each sentence`. Such customized microbenchmarks are essential for performing use-case specific benchmarking and detailed component-based testing to demonstrate their strengths and weaknesses.

To the best of our knowledge, there is no RE benchmarking framework that allows users to generate customized microbenchmarks according to user-defined criteria. Furthermore, the existing RE datasets are generally designed for specific purpose. For example, the primary purpose of the NYT-FB [18] dataset is distant supervision and is specialized for RE tasks that are based on distant supervision. Similarly, the WEB-NLG [8] dataset primarily targets natural language generation, and supervised RE systems are the main objective of the TACRED [36] dataset. The main task of the Wikidata-RE [21] dataset is to extract overlapping or multiple relationships. Consequently, no benchmark dataset is curated from multiple sources (most of the datasets have Wikipedia as a source). Riedal et al. [18] mentioned the problems caused by considering only a single source for RE systems. Finally, the RE systems reported a significant difference in the F scores for the different datasets (see Table 2).

Table 1: State-of-the-art benchmark datasets, its primary tasks and source of extraction.

Benchmark	Primary task	Underlying Corpus	Availability
NYT-FB	Distant supervision	New york times article	Partially available
Wikidata-RE	Overlaping RE	Wikipedia	Open
WEB-NLG	Natural language Generaion	Crowdsourced	Open
SEMEval-2010	RE classification	Web	Open
Google-RE	Relation Extraction	Wikipedia	Open
TACRED	Supervised RE	TAC-KBP	Closed
DocRED	Document RE	Wikipedia	Open

Table 2: Basic statistics of well-known relation extraction benchmark datasets, ^{D} represents documents instead of sentences.

Benchmark	# training Sentences	# Relation	Best F1	# NA relation
NYT-FB	561,95	24	92.5	x
Wikidata-RE	372,059	353	83	0%
WEB-NLG	501,9	246	93	0%
SEMEval-2010	10,717	9	91	17.4%
Google-RE	5528	3	87.2	0%
TACRED	106,264	41	75.2	80%
DocRED	3053^D	96	67.28	0%

Our Proposal: The performance of RE systems is significantly affected by various sentence and relations-level features, such as the number of tokens in the sentences, named entities, tokens around the mentioned entities, tokens in the entities, exact string match of the entities, and number of punctuations [26,4,22,1]. We propose REBench, an RE benchmarking framework that allows users to generate customized microbenchmarks according to user-defined criteria on various sentence and relations-level features. We use state-of-the-art clustering algorithms in REBench to cluster more representative relations and select divers microbenchmarks.

REBench selects microbenchmarks from the RELD-RDF dataset, created from six – WEB-NLG [8], NYT-FB [18], Wikidata RE [21], SemEval2010 [10], Google-RE [15], and FewRel [9], – state-of-the-art RE datasets. In RELD-RDF, we model all these datasets (which were in different formats) into a single ontology. RELD-RDF provides a unified format for data access along with various annotations which are required for training different types of relation extraction systems. The RELD-RDF resulted in the largest (to the best of our knowledge) RDF knowledge graphs of relations, containing 55.54 million triples describing 824 relations and 2 million sentences.

Our main contributions are as follows:

- REBench allows users to generate customized benchmarks according to user-defined criteria on important sentences and relation-level features. The framework completely abides by Semantic Web technologies: it uses the RDF dataset as input and makes use of SPARQL queries for sample selection and clustering.
- RELD-RDF is an assorted dataset constructed from well-known RE datasets extracted from various domains. This enables REBench to select a microbenchmark from multiple sources to avoid single-source problems [18].
- We evaluated state-of-the-art RE tools on a customized benchmark generated by REBench. The evaluation results show that baseline systems can be changed using more diverse benchmarks.

The rest of the paper is organized as follows: In Section 2, we describe the RDF dataset we used and the approach to building REBench. Section 3 presents the performance of different RE systems on the REBench. The importance and impact of the resource are explained in Section 4, and Section 5 presents resource availability, reusability and sustainability. Related work, conclusion and future work are presented in Sections 6, and 7, respectively.

2 REBench

This section first discusses the RDF dataset used as an input for the REBench relation sample generation framework. We then discuss the relation sampling process and microbenchmark generation framework in detail.

2.1 RELD-RDF Dataset

As mentioned previously, our framework selects a customized relation sample from the RELD-RDF dataset. The RELD-RDF dataset consists of six datasets that are commonly used to train and evaluate different types of RE systems. For example, WEB-NLG, NYT-FB, and Wikidata datasets are commonly used for sentential RE, Google-RE and DocRED are used for document-based RE, the FewRel dataset is used for Few-shot RE, and the SemEval2010 dataset is commonly used for casual RE[3]. In RELD-RDF, each relation contains 23 features (more than the source datasets) divided into two main categories: relation-level and sentence-level features. Features related to relations include its natural language representation; source; other representations of the relationship such as *P569, date of birth, birthDate, and /people/person/date_of_birth* all represent the same relation; and distribution (training, testing, validation). Similarly, the features related to sentences include the number of tokens, number of entities, direction of relation, position of the subject and object entity [4] in the sentence. Fig 1 summarizes the features attached to each relation. A sample RDF representation of a relationship in RELD-RDF is presented in Listing 1.1. The RELD-RDF is publicly available from the SPARQL endpoint http://reld.cs.upb.de:8890/sparql.

2.2 Relation Sample Generation for Microbenchmarking

In this section, we first define the relation sampling generation problem, followed by the generation process. We define our relation sampling generation problem as follows:

Definition 1 (Sampling problem). *Let $\mathcal{S}$ be a set of input relations. Our aim is to choose a set of $\mathcal{R}$ relations that best represents $\mathcal{S}$ with more diverse features $\mathcal{R} << |\mathcal{S}|$.*

[3] For the details about different types of RE system see Section 6.

[4] Subject and object entities sometimes also named as head and tail entities.

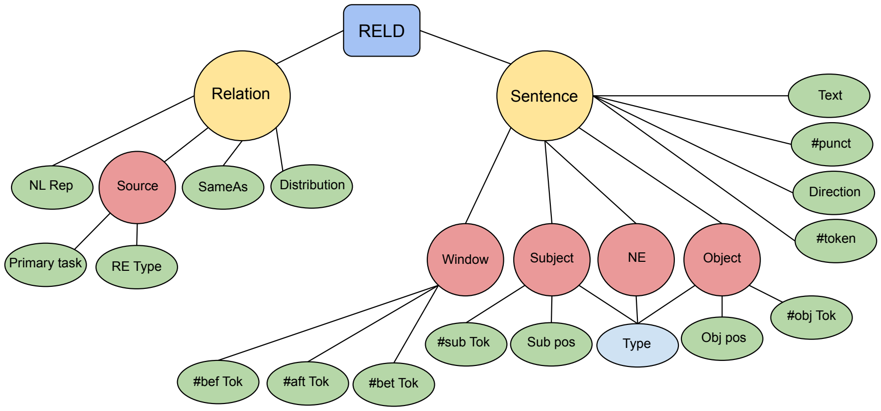

Fig. 1: A summary tree of features attached to relations and sentences in `RELD-RDF` dataset.

The relation sample generation process is carried out in four main steps, as shown in Fig 2. As a prerequisite, the user provides the `RELD-RDF` dataset as input, the required number of relation $\mathcal{R}$, and the selection criteria (as SPARQL query) to be considered in the RE sampling for microbenchmarking. The sampling process is carried out in four steps. (1) The relation selection step selects all relations with required features from the `RELD-RDF` dataset. (2) The vector representation step generates feature vectors and normalization of them for the selected relationships. (3) The model generates an $\mathcal{R}$ number of clusters from the selected relations in the clustering step. (4) Final relation selection, the model selects the most representative relation from each cluster to be included in the final sample requested by the user. We now discuss these four steps in more detail.

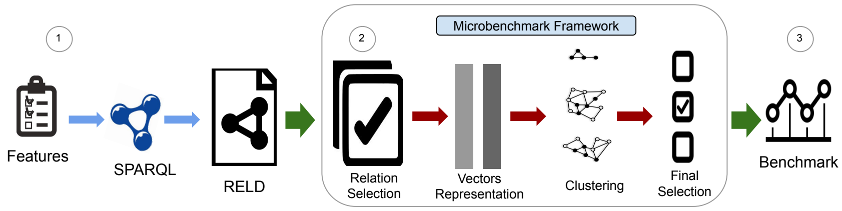

Fig. 2: `REBench` sampling process from input to output.

Listing 1.1: An example RELD-RDF representation of a relation with associated data sentences properties and associated data of sentences.

```
@prefix dataset: <https://reld.dice-research.org/> .
@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 reldr: <https://reld.dice-research.org/resource/> .
@prefix reldv: <https://reld.dice-research.org/schema/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
# Dataset #
dataset:NYT-FB reldv:hasRelation reldr:R-4001,
      reldr:Dataset_2 dc:title reldr:NYT-FB .
      reldr:NYT-FB dc:source reldr:text_freebase ;
      reldv:primaryTask reldr:distant_supervision ;
      reldv:reType reldr:ternary .
# Relation #
reldr:R-4001 rdfs:label "place_of_birth" ;
    owl:equivalentProperty reldr:placeOfBirth ;
    owl:sameAs <http://rdf.freebase.com/ns/people/person/place_of_birth>,
        ps:P19, reldr:R-2, reldr:R-3001, reldr:R-5001 ;
    reldv:distribution "test"^^xsd:string,
        "train"^^xsd:string,
        "valid"^^xsd:string ;
    reldv:hasSentence reldr:S_NYT-FB_103382,
    ...
        reldr:S_NYT-FB_106692.
# Sentence #
reldr:S_NYT-FB_103382 reldv:direction false ;
    reldv:hasNamedEntity reldr:ne_n559817, reldr:ne_n559818,
          reldr:ne_n559819, reldr:ne_n559820;
    reldv:hasObject reldr:object_55 ;
    reldv:hasSubject reldr:subject_50 ;
    reldv:hasText "Or as Heather Marks ,
    the 17-year-old Vogue favorite from Calgary , puts it : ''
    It could be that Canada is just having a moment like Brazil and
    Russia did ."@en ;
    # Sentnece Properties #
    reldv:numAftToken 21 ;
    reldv:numBefToken 2 ;
    reldv:numBetToken 6 ;
    reldv:numOfObjToken 1 ;
    reldv:numOfPunctuations 3 ;
    reldv:numOfRelation 3 ;
    reldv:numOfSubToken 2 ;
    reldv:numOfTokens 32 ;
    reldv:objPos 10 ;
    reldv:subPos 2 .
# Subject & Object #
reldr:subject_50 reldv:subject reldr:Heather_Marks .
reldr:object_55 reldv:object reldr:Calgary .
# Named Entities #
reldr:ne_n559817 a dbo:GPE ;
    rdfs:label "calgary"@en .
reldr:ne_n559818 a dbo:GPE ;
    rdfs:label "canada"@en .
reldr:ne_n559819 a dbo:GPE ;
    rdfs:label "brazil"@en .
reldr:ne_n559820 a dbo:GPE ;
    rdfs:label "russia"@en .
```

Relations Selection: There can be potentially many relations in the REDL-RDF dataset that pass the user criteria for microbenchmarks. The sampling framework fetches all relevant relations along with the required annotated features from the `RELD-RDF` dataset using a single SPARQL query. An example of a SPARQL query is presented in Listing 1.2. This SPARQL query retrieves all relations from the dataset along with the following features in the sentences: total number of tokens, named entities, and number of punctuations in a sentence. The user can select any number of features that are considered important for microbenchmarking. The result of this query execution is stored in a map that is used in subsequent sampling steps. In the following sections, we show how this query can be modified to select customized samples for microbenchmarking.

Feature Vectors:
The clustering step (explained next) requires measures of distances between relations. Each relation that was retrieved in the relation selection step from the `RELD-RDF` dataset is mapped to a vector representation. The length of the vector is equal to the number of selected features. The vector stores the corresponding relation features that were retrieved along with the given relations. Once feature vectors are created from relations, the next step is to normalize all values in the vectors between 0 and 1 to avoid bias owing to high values in the vector. The normalization of vectors for particular features is performed as follows: each of the individual values in every feature vector is divided by the overall maximal value (across all vectors) for that feature. This ensures that all the relations are located in a unit hypercube.
Clustering: Given a set of normalized vectors, the next step is to group them into the required $\mathcal{R}$ number of clusters. For this, we draw a normalized vector in the multidimensional space and used existing well-known distance-based clustering namely FEASIBLE [20], FEASIBLE Exemplars [20], KMeans++, DBSCAN+KMeans++ (Combination of DBSCAN and KMeans where DBSCAN remove outliers while KMeans generate the required number of clusters) [7], and Random selection. The `REBench` framework is not limited to these clustering methods; it is sufficiently flexible to be extended to other clustering algorithms that allow the generation of a fixed number of clusters.

Listing 1.2: SPARQL Query for selection of relations from NYT-FB from `RELD-RDF` dataset using named entity, number of punctuation and number of token features.

```
PREFIX reld: <https://reld.dice-research.org/schema/>
SELECT DISTINCT ?rId (AVG(?nToken) as ?avgToken) (count(?ne) as ?NE) (AVG(?
     numPunc) as ?avgPunc)
FROM <http://reld.dice-research.org/NYT-FB>
WHERE{
?rId reld:hasSentence ?sentence.
?sentence reld:hasSubject ?sub.
?sentence reld:hasObject ?obj.
?sentence reld:numOfTokens ?nToken.
?sentence reld:numOfPunctuations ?numPunc.
?sentence reld:hasNamedEntity ?ne.
}
```

Final Selection of Most Representative Relations: For this step, we adopt the exact approach of FEASIBLE [20] as follows: For each cluster C finds the centroid c which is the average of the feature vectors of all queries in the vectors in C. Next, we determine the distance between each relation in C and the centroid c. The final selection criterion is the minimum distance between the relationship and c. The output of our framework is an RDF file containing the selected relations, along with a list of features. This RDF output can be queried directly using a SPARQL query. The input for state-of-the-art RE systems is different, and we provide a generic script to convert the output into JSON format. The user can also convert the output into the desired style with minimum effort. `REBench` contains CLI options for benchmark generation that are available from the resource homepage.

Listing 1.3: Personalized query for selection of relations and corresponding sentences along with required features from `RELD-RDF` dataset having balanced number of sentences.

```
Prefix reld: <https://reld.dice-research.org/schema/>
SELECT  DISTINCT ?rId  (AVG(?nToken) as ?avgToken)  (AVG(?befT) as ?
    avgBeforeTokens) (AVG(?aftT) as ?avgAfterToken)
{
?rId  reld:hasSentence ?sentence.
?sentence reld:numOfTokens ?nToken.
?sentence reld:numBefToken ?befT.
?sentence reld:numAftToken ?aftT.

} Group by ?rId having (count(?sentence) = 700)
```

2.3 Relation Sample Personalization

As mentioned previously, our framework allows users to generate customized benchmarks according to user requirements. For example, a user might be interested in generating a Few-Shot (a benchmark with a balanced number of sentences for each relation) microbenchmark with 700 sentences each. To do so, the user can simply personalize the SPARQL query given in Listing 1.2 by adding SPARQL `Group By`, and `Having` clauses as shown in Listing 1.3.

2.4 Diversity of Relation Sample

Like any benchmark, the relations included in an RE benchmark should be diverse in terms of the features that affect the performance of RE systems. We define the diversity of the benchmark generated by `REBench` as follows.

Definition 2 (Sample Diversity).
Let S be a relation sample extracted from a set of relations L. The diversity score D is the average standard deviation of the relation features k included in the relation sample S:

$$D = \frac{1}{k} \sum_{i=1}^{k} (\sigma_i(S)) \qquad (1)$$

Where μ_i and σ_i represent mean and standard deviation, respectively. Where i represents the ith feature of the said distribution. In the next section, we present the diversity scores of the microbenchmarks generated using different clustering methods included in the `REBench`.

3 Evaluation and Results

This section describes the experimental setup and evaluation results.

3.1 Experimental Setup

We used three microbenchmarks in our evaluation: (1) A 15 relations sample was extracted from the `RELD-RDF` NYT-FB sub-graph to evaluate the systems trained on the NYT-FB dataset. We used the personalized query in Listing 1.2 to select these relations.

(2) To evaluate the Few-shot relation extraction model, we used listing 1.3 to extract relations with a balanced number of sentences in the `RELD-RDF` datasets. We selected 40 relations from the `RELD-RDF` dataset by keeping the number of sentences equal to 700. (3) Bootstrapping-based RE approaches are more likely to be sensitive to the features in a sentence; therefore, we choose two 100 relation benchmarks with features such as the number of tokens in a sentence, the number of tokens around the entities, and the direction property. We kept the direction property true in one benchmark and false in the second benchmark to observe the effect of direction of the entities during evaluation. We selected all these benchmarks using FEASIBLE-Exemplars because of their highest diversity score. The systems we chose for evaluation did not accept data directly in the RDF format; therefore, we converted the selected data according to the requirements of a particular RE system we chose for the evaluation.

3.2 Selected RE Systems for Evaluation

We selected those RE systems for evaluation that carry out the following criteria:

- Availability of open source implementation
- State-of-the-art baseline results
- Designed for sentence-based relation extraction

We chose three types of RE systems for the evaluation: supervised, bootstrapping, and unsupervised. Supervised systems include Partition Filter Network (PFN) [29] and Relation Extraction By End-to-end Language generation (REBEL)[11]. In addition, we selected Distributional Similarity for Relation Learning (Matching the Blanks) [2] system to evaluate on a balanced benchmark. For bootstrapping-based systems, we selected BREDS [4], and from the unsupervised category, we selected Revisiting Unsupervised Relation Extraction (URE) [26].

3.3 Results

Diversity Scores: First, we wanted to check which clustering method included in `REBench` generates more diverse benchmarks. To this end, we generated five microbenchmarks with a number of relations equal to 4, 24, 80, 200, and 350 using supported clustering methods. The diversity scores of these benchmarks are shown in Fig 3 for each supported clustering method. It is observed that FEASIBLE-Exemplars generates the most diverse benchmarks, followed by FEASIBLE, KMean++, DBSCAN+KM++, and Random selection, respectively. The reason for FEASIBLE-Exemplars high diversity is due to its clustering method: it selects exemplars based on the longest distances from each other. FEASIBLE and KMeans++ are centroid-based, instead of selecting samples based on the longest distance. The removal of outliers by DBSCAN reduced the overall diversity score. Finally, random selection does not follow a particular method for the selection of relations; therefore, its diversity score is the lowest.

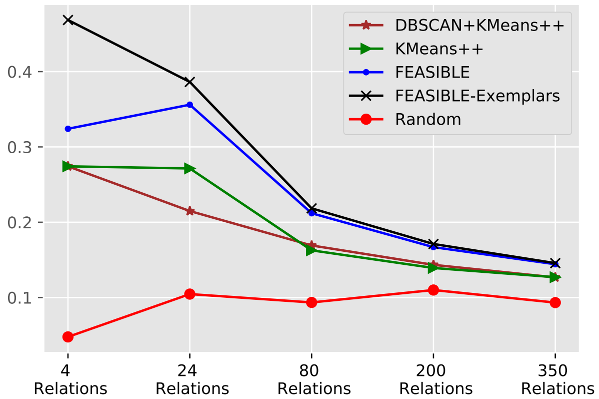

Fig. 3: Diversity score for five different algorithms using benchmarks of different size.

F measures: We now compare the performance of the selected RE systems in terms of standard precision, recall, and F measures. The evaluation results are listed in Table 3. In the supervised category, PFN slightly outperformed REBEL (in terms of F scores 92.4 vs 91.7) while using the original benchmark dataset, i.e, NYT-FB. However, REBEL clearly outperformed PFN (F score 89.9 vs 82) for `REBench`. One possible reason for the fluctuation in the results is that PFN considers a single token named entity. The results change when the number of tokens in the entity changes. This indicates that the results of an RE system depend on the diversity of the samples selected for evaluation and the different sentence and relation-level features such as the number of named entities, tokens in the sentences, can change the ranking of the tested RE systems. It is highly possible that an RE system might be tuned well for a particular type of sentence length and style, but performed worse when applied to sentences with high veracity.

Table 3: Precision, Recall and F-score of different types of RE systems on `REBench` and the original benchmark dataset, we observed fluctuation in the values and shows new baseline. * represents average F-score, while ^{F} and ^{T} represent a direction feature in a benchmark as False and True, respectively.

Type	**RE Systems**	**Dataset**	**P**	**R**	**F**
Supervised	REBEL (micro)	NYT-FB	91.5	92.0	91.7
	PFN(micro)		**92.3**	**92.5**	**92.4**
	REBEL (micro)	`REBench`	**90.4**	**89.6**	**89.9**
	PFN(micro)		84.2	80.0	82.0
Bootstrapping	BREDS	News Articles	0.79	0.80	0.79*
		`REBench` ^{F}	**0.84**	**0.87**	**0.85**
		`REBench` ^{T}	0.66	0.73	0.69
Unsupervised	URE	NYT-FB	0.31	0.63	0.41
		`REBench` ^{F}	**0.32**	**0.70**	**0.44**
		`REBench` ^{T}	0.29	0.55	0.38

Similarly, bootstrapping and unsupervised RE systems are sensitive to the structure of the sentences from which the relations are extracted. For example, our results show that simply changing the subject and object position in sentences significantly affects the F scores of the BREDS and URE RE systems. This change in results indicate the importance of customized microbenchmarks for performing diverse stress testing. Furthermore, we evaluate a Few-shot RE system [2] on listing 1.3; the overall F-score remains almost the same as that reported in the paper `(F-score = 88.9)`. The reported F-score from the original paper is based on 80 relations, while we chose the 40 most representative relations. The results indicate that our framework can select the most representative sample from the population.

4 Impact

This study provides an open source, easily extendable, and reusable resource for microbenchmarking of RE components and models. We constructed an RDF dataset from existing RE datasets, which are in different formats. We added additional features to each relation that are important to perform RE benchmarking. This dataset is publicly available and can be queried via SPARQL. The proposed dataset can be used for various NLP tasks such as relation extraction, and named entity recognition. To the best of our knowledge, no microbenchmarking framework is available for RE systems. Our proposed framework completely abides by semantic web technologies. We hope that `REBench` will be used by the NLP community to perform use-case specific benchmarking and pinpoint component-level pros and cons of RE systems.

5 Availability, Reusability and Sustainability

The resource is publicly available for reuse under the licence of GNU General Public V3.0. A detailed usage manual for reusing and adapting resources is available in our public GitHub repository. The code and usage instructions are both documented and available on the project homepage (see section 7). The resource uses Semantic Web technologies which makes its usage extendable, as well as the potential to add new clustering algorithms to the core `REBench` framework. In addition, the proposed `RELD-RDF` dataset can be extended to include more RE datasets. We provided instructions on how to reuse our code to extend the RDF dataset, as well as the `REBench` framework. All future extensions will be reflected on the same GitHub page. In addition, `REBench` will be sustained via the Paderborn Center for Parallel Computing PC^2, which provides computing resources as well as consulting regarding their usage to research projects at Paderborn University and also to external research groups. The Information and Media Technologies Centre (IMT) at Paderborn University also provides permanent IT infrastructure to host the `REBench` project.

6 Related Work

Many benchmarking datasets are available for relation extraction systems. Most of these benchmarks target a specific type of RE task. In this section, we divide them according to the target RE task.

Sentence level Relation Extraction Benchmarks: The highly explored method of relation extraction is sentence-level RE. In this type of RE, a system attempts to find the relationship between a pair of entities in a natural language sentence. A single sentence can contain one or more relations or no relation at all; similarly, a sentence can contain any number of entities. Several benchmark datasets are available for the training and evaluation of sentence-level RE systems.

NYT-FB [18]: This dataset was extracted from the New York Times and aligned to freebase [5] entities. The dataset contains 24 relations and, 56195 sentences. The dataset was initially curated for distant-supervision tasks. Some reported shortcomings of this dataset are that the dataset does not contain overlapping sentences [35], it suffers from the problem of long-tailed distribution of sentences and imbalanced relations in terms of sentence annotation [23], and Wang et al. [27] found a problem related to NER format and only the last word annotation, which directly affects the performance. Wei et al. [28] reported that

a single relation annotation in NYT-FB degrades the overall performance. *TACRED* [35]: is a well-known benchmark dataset for RE systems. The datasets contained 41 relations, which also include NA (no relation). The dataset is not available as open source. Sample imbalance, a high noise rate, and incorrect annotations have been reported in TACRED [13,16,37].
Wikipedia-Wikidata-RE: This is a comparatively large dataset in terms of relations and number of sentences. Sentences were extracted from Wikipedia and aligned to the entities of Wikidata. The dataset contains 353 relations and 372,059 training sentences. There is a high difference in the macro and micro evaluation on these datasets [3] Furthermore, some relations are sparse [38] that significantly affect the overall performance.
WEB-NLG: A natural language generation dataset containing 5019 crowdsourced training sentences and 246 relations. It is a widely used dataset for RE and has achieved human-level accuracy. Researchers have identified multiple problems regarding this dataset such as long-tail distribution, last word annotation, confusing relation labels, noisy sentences, and issue related to NER [23,27,24,32].

Apart from these benchmark datasets, there are other datasets which target sentence-level RE, such as SciERC [12], Trex [6] and, CoNLL2004 [19].
Document level Relation Extraction Benchmarks:
A relation in natural language may or may not explicitly exist in a single sentence, but comes from the context of other surrounding sentences. Therefore, sentence-level relation benchmarks do not fulfil this requirement. Document-level relation extraction benchmarks like DocRED [31] and Google-RE [15] are used for this purpose [34]. One of the main disadvantages of these benchmarks is that the source of the sentences is mostly Wikipedia. The Google-RE dataset only contains four relations and the primary task is not document-based relation extraction, while DocRED consists of 96 relations.
Causal Relation Extraction Benchmarks:
A relationship between two entities e1 and e2, such that the occurrence of e1 results in the occurrence of e2, is known as a cause–effect relation or causal relation extraction [30]. SemEval 2010 Task 8 and TACRED, contain causality relationships (1331 in SemEval 2010 Task 8 and 269 in TACRED). The main disadvantage of these benchmarks is the size (in terms of number of sentences) of the benchmarks.

None of the above benchmarking datasets provide a customized microbenchmark; neither of them uses Semantic Web technologies such as SPARQL querying. Our proposed framework `REBench` overcomes these problems and provides task specific, component-level microbenchmarks according to the user requirements.

7 Conclusion and Future Work

In this article, we describe a resource for generating samples of relations and sentences for microbenchmarking of relation extraction systems. Our resource uses different clustering algorithms to create more diverse clusters of samples to evaluate the relation extraction task. Users can select personalized samples for microbenchmarks based on the required features. The results indicate that diversity in the benchmark sample is key to performing fine-grained evaluations of RE systems. Microbenchmarking is key to performing such fine-grained component-level performance evaluations. Using our resources, the NLP community can evaluate their relation extraction systems based on their specific needs. We aim to extend our work to other natural language processing tasks, such as named entity disambiguation and Named Entity recognition.

Resource Availability Statement:

- Source code, usage instruction, evaluation results, JSON conversion code and code for generation of Fig 3 of `REBench` is available from our GitHub repository[5]
- Details about the `RELD-RDF` dataset is available on the GitHub Repository[6]
- Online endpoint of the data used in `REBench` is available on[7]

Acknowledgments

This work has been supported by the BMWK-funded project RAKI (01MD19012B), SPEAKER (01MK20011U), BMBF-funded EuroStars project PORQUE (01QE2056C), 3DFed (01QE2114B) and partially supported by DFG within the Collaborative Research Centre SFB 901 (160364472) and the University of Malakand Pakistan.

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. Baldini Soares, L., FitzGerald, N., Ling, J., Kwiatkowski, T.: Matching the blanks: Distributional similarity for relation learning. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2895–2905. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1279
3. Bastos, A., Nadgeri, A., Singh, K., Mulang, I.O., Shekarpour, S., Hoffart, J., Kaul, M.: Recon: relation extraction using knowledge graph context in a graph neural network. In: Proceedings of the Web Conference 2021. pp. 1673–1685 (2021)
4. Batista, D.S., Martins, B., Silva, M.J.: Semi-supervised bootstrapping of relationship extractors with distributional semantics. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 499–504. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015). https://doi.org/10.18653/v1/D15-1056

[5] https://github.com/dice-group/REBench
[6] https://github.com/dice-group/RELD
[7] http://reld.cs.upb.de:8890/sparql

5. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. pp. 1247–1250 (2008)
6. 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). European Language Resources Association (ELRA), Miyazaki, Japan (May 2018)
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd. vol. 96, pp. 226–231 (1996)
8. 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
9. 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: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4803–4809. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018). https://doi.org/10.18653/v1/D18-1514
10. 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)
11. Huguet Cabot, P.L., Navigli, R.: REBEL: Relation extraction by end-to-end language generation. In: Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 2370–2381. Association for Computational Linguistics, Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.findings-emnlp.204
12. Luan, Y., He, L., Ostendorf, M., Hajishirzi, H.: Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3219–3232. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018). https://doi.org/10.18653/v1/D18-1360
13. Lyu, S., Chen, H.: Relation classification with entity type restriction. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 390–395. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.findings-acl.34
14. Ning, Q., Feng, Z., Roth, D.: A structured learning approach to temporal relation extraction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 1027–1037. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017). https://doi.org/10.18653/v1/D17-1108
15. Orr, D.: 50,000 lessons on how to read: a relation extraction corpus. Online: Google Research Blog **11** (2013)
16. Park, S., Kim, H.: Improving Sentence-Level Relation Extraction through Curriculum Learning. arXiv e-prints arXiv:2107.09332 (Jul 2021)

17. Pawar, S., Palshikar, G.K., Bhattacharyya, P.: Relation extraction: A survey. arXiv preprint arXiv:1712.05191 (2017)
18. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 148–163. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
19. Roth, D., Yih, W.t.: A linear programming formulation for global inference in natural language tasks. In: Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004. pp. 1–8. Association for Computational Linguistics, Boston, Massachusetts, USA (May 6 - May 7 2004)
20. Saleem, M., Mehmood, Q., Ngonga Ngomo, A.C.: Feasible: A feature-based sparql benchmark generation framework. In: Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K., Thirunarayan, K., Staab, S. (eds.) The Semantic Web - ISWC 2015. pp. 52–69. Springer International Publishing, Cham (2015)
21. 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
22. Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Van Ess-Dykema, C., Meteer, M.: Dialogue act modeling for automatic tagging and recognition of conversational speech. Computational Linguistics **26**(3), 339–374 (2000)
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. Sun, K., Zhang, R., Mensah, S., Mao, Y., Liu, X.: Recurrent interaction network for jointly extracting entities and classifying relations. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 3722–3732. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.304
25. 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. Association for Computational Linguistics, Jeju Island, Korea (Jul 2012)
26. 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://doi.org/10.18653/v1/2020.acl-main.669
27. 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
28. Wei, Z., Su, J., Wang, Y., Tian, Y., Chang, Y.: A novel cascade binary tagging framework for relational triple extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1476–1488. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.136

29. Yan, Z., Zhang, C., Fu, J., Zhang, Q., Wei, Z.: A partition filter network for joint entity and relation extraction. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 185–197. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.17
30. Yang, J., Han, S.C., Poon, J.: A survey on extraction of causal relations from natural language text. Knowl. Inf. Syst. **64**(5), 1161–1186 (may 2022). https://doi.org/10.1007/s10115-022-01665-w
31. 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. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 764–777. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1074
32. Ye, H., Zhang, N., Deng, S., Chen, M., Tan, C., Huang, F., Chen, H.: Contrastive triple extraction with generative transformer (2020). https://doi.org/10.48550/ARXIV.2009.06207
33. Yu, M., Yin, W., Hasan, K.S., dos Santos, C., Xiang, B., Zhou, B.: Improved neural relation detection for knowledge base question answering. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 571–581. Association for Computational Linguistics, Vancouver, Canada (Jul 2017). https://doi.org/10.18653/v1/P17-1053
34. Zaporojets, K., Deleu, J., Develder, C., Demeester, T.: DWIE: an entity-centric dataset for multi-task document-level information extraction. Information Processing & Management **58**(4), 102563 (2021). https://doi.org/https://doi.org/10.1016/j.ipm.2021.102563
35. Zeng, X., Zeng, D., He, S., Liu, K., Zhao, J.: Extracting relational facts by an end-to-end neural model with copy mechanism. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 506–514. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). https://doi.org/10.18653/v1/P18-1047
36. 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. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017). https://doi.org/10.18653/v1/D17-1004
37. Zhou, W., Chen, M.: An improved baseline for sentence-level relation extraction (2021). https://doi.org/10.48550/ARXIV.2102.01373
38. Zhu, H., Lin, Y., Liu, Z., Fu, J., Chua, T.S., Sun, M.: Graph neural networks with generated parameters for relation extraction. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 1331–1339. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1128