BENGAL: An Automatic Benchmark Generator for Entity Recognition and Linking

Axel-Cyrille Ngoma Ngomo[1,2] **Michael Röder**[1] **Diego Moussallem**[1,2] **Ricardo Usbeck**[1] **René Speck**[1,2]
[1]Data Science Group, University of Paderborn, Germany
[2]AKSW Research Group, University of Leipzig, Germany
{first.lastname}@upb.de
lastname@informatik.uni-leipzig.de

Abstract

The manual creation of gold standards for named entity recognition and entity linking is time- and resource-intensive. Moreover, recent works show that such gold standards contain a large proportion of mistakes in addition to being difficult to maintain. We hence present BENGAL, a novel automatic generation of such gold standards as a complement to manually created benchmarks. The main advantage of our benchmarks is that they can be readily generated at any time. They are also cost-effective while being guaranteed to be free of annotation errors. We compare the performance of 11 tools on benchmarks in English generated by BENGAL and on 16 benchmarks created manually. We show that our approach can be ported easily across languages by presenting results achieved by 4 tools on both Brazilian Portuguese and Spanish. Overall, our results suggest that our automatic benchmark generation approach can create varied benchmarks that have characteristics similar to those of existing benchmarks. Our approach is open-source. Our experimental results are available at http://faturl.com/bengalexpinlg and the code at https://github.com/dice-group/BENGAL.

1 Introduction

The creating of gold standard is of central importance for the objective assessment and development of approaches all around computer science. For example, evaluation campaigns such as BioASQ (Tsatsaronis et al., 2012) have led to an improvement of the F-measure achieved by biomedical question answering systems by more than 5%. While the manual creation of Named Entity Recognition (NER) and Entity Linking (EL) gold standards (also called benchmarks) has the advantage of yielding resources which reflect human processing, it also exhibits significant disadvantages: **a)** *Annotation mistakes*: Human annotators have to read through every sentence in the corpus and often (a) miss annotations or (b) assign wrong resources to entities for reasons as various as fatigue or lack of background knowledge (and this even when supported with annotation tools). For example, Jha et al. (2017) was able to determine that up to 38,453 of the annotations in commonly used benchmarks (see GERBIL (Usbeck et al., 2015) for a list of these benchmarks) were erroneous. A manual evaluation of 25 documents from the ACE2004 benchmark revealed that 195 annotations were missing and 14 of 306 annotations were incorrect. Similar findings were reported for AIDA/CONLL (Tjong Kim Sang and De Meulder, 2003) and OKE2015 (Nuzzolese et al., 2015). **b)** *Volume*: Manually created benchmarks are usually small (commonly $< 2,500$ documents, see Table 2). Hence, they are of little help when aiming to benchmark the scalability of existing solutions (especially when these solutions use caching). **c)** *Lack of updates*: Manual benchmark generation approaches lead to static corpora which tend not to reflect the newest reference knowledge graphs (also called Knowledge Base (KB)s). For example, several of the benchmarks presented in GERBIL (Usbeck et al., 2015) link to outdated versions of Wikipedia or DBpedia. **d)** *Popularity bias*: van Erp et al. (2016) show that manual benchmarks are often biased towards popular resources. **e)** *Lack of availability*: The lack of benchmarks for resource-poor languages inhibits the development of corresponding

Proceedings of The 11th International Natural Language Generation Conference, pages 339–349,
Tilburg, The Netherlands, November 5-8, 2018. ©2018 Association for Computational Linguistics

NER and EL solutions.

Automatic methods are a *viable and supplementary* approach for the generation of gold standards for NER and EL, especially as they address some of the weaknesses of the manual benchmark creation process. *The main contribution of our paper is a novel approach for the automatic generation of benchmarks for NER and EL* dubbed BENGAL. Our approach relies on the abundance of structured data in Resource Description Framework (RDF) on the Web and is based on Natural Language Generation (NLG) techniques which verbalize such data to generate automatically annotated natural language statements. Our automatic benchmark creation method addresses the drawbacks of manual benchmark generation aforementioned as follows: **a)** It alleviates the human annotation error problem by relying on data in RDF which explicitly contain the entities to find. **b)** BENGAL is able to generate arbitrarily large benchmarks. Hence, it can enhance the measurement of both the accuracy and the scalability of approaches. **c)** BENGAL can be updated easily to reflect the newest terminology and reference KBs. Hence, it can generate corpora that reflect the newest KBs. **d)** BENGAL is not biased towards popular resources as it can choose entities to include in the benchmark generated following a uniform distribution. **e)** BENGAL can be ported to any token-based language. This is exemplified by porting BENGAL to Portuguese and Spanish.

2 Related Work

2.1 Gold Standards for NER and EL

According to GERBIL (Usbeck et al., 2015), the 2003 CoNLL shared task (Tjong Kim Sang and De Meulder, 2003) is the most used benchmark dataset for recognition and linking. The ACE2004 and MSNBC (Cucerzan, 2007) news datasets were used by Ratinov et al. (Ratinov et al., 2011) to evaluate their seminal work on linking to Wikipedia. Another often-used corpus is AQUAINT, e.g., used by Milne and Witten (Milne and Witten, 2008). Detailed dataset statistics on some of these benchmarks can be found in Table 2.

A recent uptake of publicly available corpora (Röder et al., 2014; Steinmetz et al., 2013) based on RDF has led to the creation of many new datasets. For example, the Spotlight corpus and the KORE 50 dataset were proposed to showcase the usability of RDF-based annotations (Mendes et al., 2011). The multilingual N3 collection (Röder et al., 2014) was introduced to widen the scope and diversity of NIF-based corpora. Another recent observation is the shift towards gold standards for micropost documents like tweets. For example, the Microposts2014 corpus (Cano Basave et al., 2014) was created to evaluate NER on smaller pieces of text.

Semi-automatic approaches to benchmark creation are commonly crowd-based. They use one or more recognizers to create a first set of annotations and then hand over the tasks of refinement and/or linking to crowd workers to improve the quality. Examples of such approaches include Voyer et al. (2010) and CALBC (Rebholz-Schuhmann et al., 2010). Oramas et al. (2016) introduced a voting-based algorithm which analyses the hyperlinks presented in the input texts retrieved from different disambiguation systems such as Babelfy (Moro et al., 2014). Each entity mention in the input text is linked based on the degree of agreement across three EL systems.

BENGAL is the first automatic approach that makes use of structured data and can be replicated on any RDF KB for EL benchmarks.

2.2 NLG for the Web of Data

A plethora of works have investigated the generation of Natural Language (NL) texts from Semantic Web Technologies (SWT) such as Staykova (2014); Bouayad-Agha et al. (2014). However, the generation of NL from RDF has only recently gained momentum. This attention comes from the great number of published works such as (Cimiano et al., 2013; Duma and Klein, 2013; Ell and Harth, 2014; Biran and McKeown, 2015) which used RDF as an input data and achieved promising results. Moreover, the works published in the WebNLG (Colin et al., 2016) challenge, which used deep learning techniques such as (Sleimi and Gardent, 2016; Mrabet et al., 2016), also contributed to this interest. RDF has also been showing promising benefits to the generation of benchmarks for evaluating NLG systems, e.g., (Gardent et al., 2017; Perez-Beltrachini et al., 2016; Mohammed et al., 2016; Schwitter et al., 2004; Hewlett et al., 2005; Sun and Mellish, 2006). However, RDF has never been used for creating NER and NEL benchmarks. BENGAL addresses this research gap.

3 The BENGAL approach

BENGAL is based on the observation that more than 150 billion facts pertaining to more than 3 billion entities are available in machine-readable form on the Web (i.e., as RDF triples).[1] The basic intuition behind our approach is hence as follows: *Given that NER and EL are often used in pipelines for the extraction of machine-readable facts from text, we can invert the pipeline and go from facts to text*, thereby using the information in the facts to produce a gold standard that is *guaranteed to contain no errors*. In the following, we begin by giving a brief formal overview of RDF. Thereafter, we present how we use RDF to generate NER and EL benchmarks automatically and at scale.

3.1 Preliminaries and Notation

3.1.1 RDF

The notation presented herein is based on the RDF 1.1 specification. An RDF graph G is a set of facts. Each fact is a triple $t = (s, p, o) \in (R \cup B) \times P \times (R \cup B \cup L)$ where R is the set of all resources (i.e., things of the real world), P is the set of all predicates (binary relations), B is the set of all blank nodes (which basically express existential quantification) and L is the set of all literals (i.e., of datatype values). We call the set $R \cup P \cup L \cup B$ our universe and call its elements entities. A fragment of DBpedia[2] is shown below. We will use this fragment in our examples. For the sake of space, our examples are in English. However, note that we ported BENGAL to Portuguese and Spanish so as to exemplify that it is not biased towards a particular language. Also, the morphological richness of both led us to choose them as languages.

```
:Albert_Einstein dbo:birthPlace :Ulm .
:Albert_Einstein dbo:deathPlace :
    Princeton .
:Albert_Einstein rdf:type dbo:Scientist
    .
:Albert_Einstein dbo:field :Physics .
:Ulm dbo:country :Germany.
:Albert_Einstein rdfs:label "Albert␣
    Einstein"@en.
```

Listing 1: Example RDF dataset.

3.1.2 Benchmarks

We define a benchmark as a set C of annotated documents D_i. Each document D_i is a sequence of characters $s_{i1} \dots s_{in}$. Each subsequence $s_{ij} \dots s_{ik}$ (with $j < k$) of the document D_i which stands for a resource $r \in R$ is assumed to be marked as such. We model the marking of resources by the function $m : C \times \mathbb{N} \times \mathbb{N} \rightarrow R$ and write $m(D_i, j, k) = r$ to signify that the substring $s_{ij} \dots s_{ik}$ stands for the resource r. In case the substring $s_{ij} \dots s_{ik}$ does not stand for a resource, we write $m(D_i, j, k) = \epsilon$. Let D_0 be the example shown in Listing 2. We would write $m(D_0, 0, 14) =$:AlbertEinstein.

```
Albert Einstein was born in Ulm.
```

Listing 2: Example sentence.

3.2 Verbalization

The notation and formal framework for verbalization in BENGAL are based on SPARQL2NL (Ngonga Ngomo et al., 2013). Let W be the set of all words in the dictionary of our target language (e.g., English). We define the realization function $\rho : R \cup P \cup L \rightarrow W^*$ as the function which maps each entity to a word or sequence of words from the dictionary. Formally, the goal of our NLG approach is to devise an extension of ρ to conjunctions of RDF triples. This extension maps all triples t to their realization $\rho(t)$ and defines how these atomic realizations are to be combined. We denote the extension of ρ by the same label ρ for the sake of simplicity. We adopt a rule-based approach to devise the extension of ρ, where the rules extending ρ to RDF triples are expressed in a conjunctive manner. This means that for premises $P_1, \dots, P_n$ and consequences $K_1, \dots, K_m$ we write $P_1 \wedge \dots \wedge P_n \Rightarrow K_1 \wedge \dots \wedge K_m$. The premises and consequences are explicated by using an extension of the Stanford dependencies.[3] We rely especially on the constructs explained in Table 1. For example, a possessive dependency between two phrase elements e_1 and e_2 is represented as $\texttt{poss}(e_1, e_2)$. For the sake of simplicity, we sometimes reduce the construct $\texttt{subj(y,x)} \wedge \texttt{dobj(y,z)}$ to the triple $\texttt{(x,y,z)} \in W^3$.

3.3 Approach

BENGAL assumes that it is given (1) an RDF graph $G \subseteq (R \cup B) \times P \times (R \cup B \cup L)$, (2) a number of

[1] http://stats.lod2.eu

[2] http://dbpedia.org

[3] For a complete description of the vocabulary, see http://nlp.stanford.edu/software/dependencies_manual.pdf.

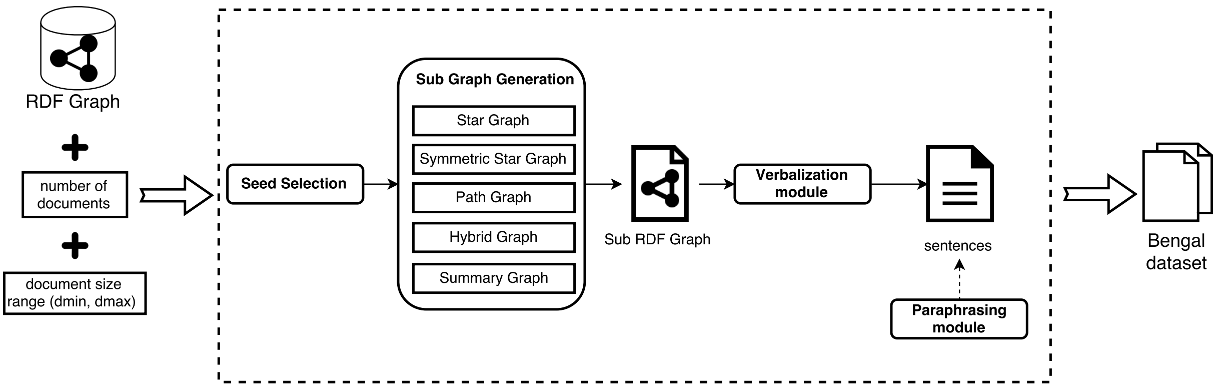

Figure 1: Overview of the BENGAL approach.

Table 1: Dependencies used by BENGAL.

Dependency	Explanation
`cc`	Stands for the relation between a conjunct and a given conjunction (in most cases and or or). For example in the sentence John eats an apple and a pear, `cc(PEAR,AND)` holds. We mainly use this construct to specify reduction and replacement rules.
`conj`*	Used to build the *conjunction* of two phrase elements, e.g. `conj(subj(EAT,JOHN), subj(DRINK,MARY))` stands for John eats and Mary drinks. `conj` is not to be confused with the logical conjunction $\wedge$, which we use to state that two dependencies hold in the same sentence. For example `subj(EAT,JOHN)` $\wedge$ `dobj(EAT,FISH)` is to be read as John eats fish.
`dobj`	Dependency between a verb and its *direct object*, for example `dobj(EAT,APPLE)` expresses to eat an/the apple.
`nn`	The *noun compound modifier* is used to modify a head noun by the means of another noun. For instance `nn(FARMER,JOHN)` stands for farmer John.
`poss`	Expresses a possessive dependency between two lexical items, for example `poss(JOHN,DOG)` expresses John's dog.
`subj`	Relation between *subject* and verb, for example `subj(BE,JOHN)` expresses John is.

documents to generate, (3) a minimal resp. maximal document size (i.e., number of triples to use during the generation process) d_{min} resp. d_{max}, (4) a set of restrictions pertaining to the resources to generate and (5) a strategy for generating single documents. Given the graph G, BENGAL begins by selecting a set of *seed resources* from G based on the restrictions set using parameter (4). Thereafter, it uses the strategy defined via parameter (5) to select a subgraph of G. This subgraph contains a randomly selected number d of triples with $d_{min} \leq d \leq d_{max}$. The subgraph is then verbalized. The verbalization is annotated automatically and finally returned as a single document. Each single document then may be paraphrased if this option is chosen in the initial phase. This process is repeated as many times as necessary to reach the predefined number of documents. In the following, we present the details of each step underlying our benchmark generation process displayed in Figure 1.

3.3.1 Seed Selection

Given that we rely on RDF, we model the seed selection by means of a SPARQL SELECT query with one projection variable. Note that we can use the wealth of SPARQL to devise seed selection strategies of arbitrary complexity. However, given that NER and EL frameworks commonly focus on particular classes of resources, we are confronted with the condition that the seeds must be instances of a set of classes, e.g., `:Person`, `:Organization` or `:Place`. The SPARQL query for our example dataset would be as follows:

```
SELECT ?x WHERE { {?x a :Person.} UNION
    {?x a :Organization.} UNION {?x a :
    Place.} }
```

Listing 3: Example seed selection query.

3.3.2 Subgraph Generation

Our approach to generating subgraphs is reminiscent of SPARQL query topologies as available in SPARQL query benchmarks. As these queries (e.g., FEASIBLE[4] queries) describe real informa-

[4] http://aksw.org/Projects/Feasible

tion needs, their topology must stand for the type of information that is necessitated by applications and humans. We thus distinguish between three main types of subgraphs to be generated from RDF data: (1) *star graphs* provide information about a particular entity (e.g, the short biography of a person); (2) *path graphs* describe the relations between two entities (e.g., the relation between a gene and a side-effect); (3) *hybrid graphs* are a mix of both and commonly describe a specialized subject matter involving several actors (e.g., a description of the cast of a movie).

Star Graphs. For each $s_i \in S$, we gather all triples of the form $t = (s_i, p, o) \in R \times P \times (R \cup L)$.[5] The triples are then added to a list $L(s_i)$ sorted in descending order according to a hash function h. After randomly selecting a document size d between d_{min} and d_{max}, we select d random triples from $L(s_i)$. For the dataset shown in Listing 1 and $d = 2$, we would for example get Listing 4.

```
:AlbertEinstein :birthPlace :Ulm .
:AlbertEinstein :deathPlace :Princeton .
```

Listing 4: Example dataset generated by the star strategy.

Symmetric Star Graphs. As above with $t \in \{(s_i, p, o) \in G \vee (o, p, s_i) \in G\}$.

Path Graphs. For each $s_i \in S$, we begin by computing list $L(s_i)$ as in the symmetric star graph generation. Then, we pick a random triple (s_i, p, o) or (o, p, s_i) from $L(s_i)$ that is such that o is a resource. We then use o as seed and repeat the operation until we have generated d triples, where d is randomly generated as above. For the example dataset shown in Listing 1 and $d = 2$, we would for example get Listing 5.

```
:AlbertEinstein :birthPlace :Ulm .
:Ulm :country :Germany .
```

Listing 5: Example dataset generated by the path strategy.

Hybrid Graphs. This is a 50/50-mix of the star and path graph generation approaches. In each iteration, we choose and apply one of the two strategies above randomly. For example, the hybrid graph generation can generate:

```
:AlbertEinstein :birthPlace :Ulm .
:AlbertEinstein :deathPlace :Princeton .
:Ulm :country :Germany .
```

Listing 6: Example dataset generated by the hybrid strategy.

Summary Graph Generation. This last strategy is a specialization of the star graph generation where the set of triples to a resource is not chosen randomly. Instead, for each class (e.g., `:Person`) of the input KB, we begin by filtering the set of properties and only consider properties that (1) have the said class as domain and (2) achieve a coverage above a user-set threshold (60% in our experiments) (e.g., `:birthPlace`, `:deathPlace`, `:spouse`). We then build a property co-occurence graph for the said class in which the nodes are the properties selected in the preceding step and the co-occurence of two properties p_1 and p_2 is the instance r of the input class where $\exists o_1, o_2 : (r, p_1, o_1) \in K \wedge (r, p_2, o_2) \in K$. The resulting graph is then clustered (e.g., by using the approach presented by Ngonga Ngomo and Schumacher (2009)). We finally select the clusters which contain the properties with the highest frequencies in K that allow the selection of at least d triples from K. For example, if `:birthPlace` (frequency = 10), `:deathPlace` (frequency = 10) were in the same cluster while `:spouse` (frequency = 8) were in its own cluster, we would choose the pair (`:birthPlace`, `:deathPlace`) and return the corresponding triples for our input resource. Hence, we would return Listing 4 for our running example.

3.3.3 Verbalization module

The verbalization (micro-planning) strategy for the first four strategies consists of verbalizing each triple as a single sentence and is derived from SPARQL2NL (Ngonga Ngomo et al., 2013). To verbalize the subject of the triple $t = (s, p, o)$, we use one of its labels according to Ell et al. (2011) (e.g., the `rdfs:label`). If the object o is a resource, we follow the same approach as for the subject. Importantly, the verbalization of a triple $t = (s, p, o)$ depends mostly on the verbalization of the predicate `p` (see Table 1 for semantics). If `p` can be realized as a noun phrase, then a possessive clause can be used to express the semantics of (s, p, o). For example, if p can be verbalized as a nominal compound like `birth place`, then the verbalization $\rho(s, p, o)$ of the triple is as follows: `poss`($\rho(p)$,$\rho(s)$) $\wedge$

[5] Note that we do not consider blank nodes as they cannot be verbalized due to the existential quantification they stand for.

$\texttt{subj}(\texttt{BE},\rho(p)) \wedge \texttt{dobj}(\texttt{BE},\rho(o))$. In case p's realization is a verb, then the triple can be verbalized as $\texttt{subj}(\rho(p),\rho(s)) \wedge \texttt{dobj}(\rho(p),\rho(o))$. In our example, verbalizing (`:AlbertEinstein`, `dbo:birthPlace`, `:Ulm`) would thus lead to `Albert Einstein's birth place is Ulm.`, as `birth place` is a noun.

In the case of summary graphs, we go beyond the verbalization of single sentences and merge sentences that were derived from the same cluster. For example, if p_1 and p_2 can be verbalized as nouns, then we apply the following rule: $\rho(s, p_1, o_1) \wedge \rho(s, p_2, o_2) \Rightarrow \texttt{conj}(\texttt{poss}(\rho(p_1), \rho(s)) \wedge \texttt{subj}(\texttt{BE}_1, \rho(p_1)) \wedge \texttt{dobj}(\texttt{BE}_1, \rho(o_1)) \wedge \texttt{poss}(\rho(p_2), \rho(\texttt{pronoun}(s))) \wedge \texttt{subj}(\texttt{BE}_2, \rho(p_2)) \wedge \texttt{dobj}(\texttt{BE}_2, \rho(o_2))$. Note that `pronoun(s)` returns the correct pronoun for a resource based on its type and gender. Therewith, we can generate `Albert Einstein's birth place is Ulm and his death place is Princeton`.

3.3.4 Paraphrasing

With this step, BENGAL avoids the generation of a large number of sentences that share the same terms and the same structure. Additionally, this step makes the use of reverse engineering strategies for the generation more difficult as it increases the diversity of the text in the benchmarks. Our paraphrasing is largely based on Androutsopoulos and Malakasiotis (2010) and runs as follows:

1. Change the structure of the sentence: We use the location of verbs in each sentence to randomly change passive into active structures and vice-versa. Sentences which describe type information (e.g., `Einstein is a person`) are not altered.

2. Replace synonyms: We use POS tags to select alternative labels from the knowledge base and a reference dictionary to replace entity labels by a synonym.

An example of a paraphrase generated by BENGAL is shown in Listing 7.

```
Original: Edmund Pettus Bridge is a
    bridge. It crosses Alabama River.
    Its type is Through arch bridge. It
    was declared a National Historic
    Landmark on March 11, 2013.

Paraphrased: Edmund Pettus Bridge is a
    bridge. It crosses Alabama River.
    Through arch bridge is its type.
    Pettus was declared a National
    Historic Landmark on March 11, 2013.
```

Listing 7: Example Paraphasing at Summary Generation

4 Experiments and Results

We generated 13 datasets in English (B1-B13), 4 datasets in Brazilian Portuguese and 4 datasets in Spanish to evaluate our approach.[6] B1 to B10 were generated by running our five sub-graph generation methods with and without paraphrasing. The number of documents was set to 100 while (d_{min}, d_{max}) was set to $(1, 5)$. B11 shows how BENGAL can be used to evaluate the scalability of approaches.[7] Here, we used the hybrid generation strategy to generate 10,000 documents. B12 and B13 comprise 10 longer documents each with d_{min} set to 90. For B12, we focused on generating a high number of entities in the documents while B13 contains less entities but the same number of documents.

We compared B1-B13 with the 16 manually created gold standards for English found in GERBIL. The comparison was carried out in two ways. First, we assessed the features of the datasets. Then, we compared the micro F-measure of 11 NER and EL frameworks on the manually and automatically generated datasets. We chose to use these 11 frameworks because they are included in GERBIL. This inclusion ensures that their interfaces are compatible and their results comparable. In addition, we assessed the performance of multilingual NER and EL systems on the datasets P1-P4 to show that BENGAL can be easily ported to languages other than English.

4.1 English Dataset features

The first aim of our evaluation was to quantify the variability of the datasets B1–B13 generated by BENGAL. To this end, we compared the distribution of the part of speech (POS) tags of the BENGAL datasets with those of the 16 benchmark datasets. An analysis of the Pearson correlation of these distributions revealed that the manually

[6] All BENGAL datasets can be found at https://hobbitdata.informatik.uni-leipzig.de/bengal/

[7] The scalability results are available at https://goo.gl/9mnbwC and cannot be presented herein due to space limitations.

created datasets (D1–D16) have a high correlation (0.88 on average) with a minimum of 0.61 (D10–D16). The correlation of the POS tag distributions between BENGAL datasets and a manually created dataset vary between 0.34 (D7–B11) and 0.89 (D14–B9) with an average of 0.67. This shows that BENGAL datasets can be generated to be similar to manually created datasets (D14–B9) as well as to be very different to them (D7–B11). Hence, BENGAL can be used for testing sentence structures that are not common in the current manually generated benchmarks.[8]

We also studied the distribution of entities and tokens across the datasets in our evaluation. Table 2 gives an overview of these distributions, where E is the set of entities in the corpus C. The distribution of values for the different features is very diverse across the different manually created datasets. This is mainly due to (1) different ways to annotate entities and (2) the domains of the datasets (news, description of entities, microposts). As shown in Table 2, BENGAL can be easily configured to generate a wide variety of datasets with similar quality and number of documents to those of real datasets. This is mainly due to our approach being able to generate benchmarks ranging from (1) benchmarks with sentences containing a large number of entities without any filler terms (high entity density) to (2) benchmarks which contain more information pertaining to entity types and literals (low entity density).

4.2 Annotator performance

We used GERBIL to evaluate the performance of 11 annotators on the manually created as well as the BENGAL datasets. We evaluated the annotators within an A2KB (annotation to knowledge base) experiment setting: Each document of the corpora was sent to each annotator. The annotator had to find and link all entities to a reference KB (here DBpedia). We measured both the performance of the NER and the EL steps.

Table 3 shows the micro F1-score of the different annotators on chosen datasets. The manually created datasets showed diverse results. We analyzed the results further by using the F1-scores of the annotators as features of the datasets. Based on these feature vectors, we calculated the Pearson correlations between the datasets to identify datasets with similar characteristics.[9] The Pearson correlations of the F-measures achieved by the different annotators on the AIDA/CoNLL datasets (D2–D5) are very high (0.95–1.00) while the correlation between the results on the Spotlight corpus (D7) and N3-Reuters-128 (D13) is around -0.62. The results on D1 and D12–D15 have a correlation to the AIDA/CoNLL results (D2–D5) that is higher than 0.5. In contrast, the correlations of D7 and D8 to the AIDA/CoNLL datasets range from -0.54 to -0.36. These correlations highlight the diversity of the manually created datasets and suggest that creating an approach which emulates all datasets is non-trivial.

Like the correlations between the manually created datasets, the correlations between the results achieved on BENGAL datasets and hand-crafted datasets vary. The results on BENGAL correlate most with the results on the OKE 2015 data. The highest correlations were achieved with the OKE 2015 Task 1 dataset and range between 0.89 and 0.92. This suggests that our benchmark can emulate entity-centric benchmarks. The correlation of BENGAL with OKE is however reduced to 0.82 in D13, suggesting that BENGAL can be parametrized so as to diverge from such benchmarks. A similar observation can be made for the correlation D12 and ACE2004, where the correlation increased with the size of the documents in the benchmark. The correlation between the results across BENGAL datasets varies between 0.54 and 1, which further supports that BENGAL can generate a wide range of diverse datasets.

4.3 Annotator Performance on Spanish and Brazilian Portuguese

We implemented BENGAL for Brazilian Portuguese by using the RDF verbalizer presented in Moussallem et al. (2018) and ran four multilingual NER and EL (MAG (Moussallem et al., 2017), DBpedia Spotlight, Babelfy, and PBOH (Ganea et al., 2016)) frameworks thereon. We also evaluated the performance of these annotators on subsets of the HAREM datasets (Freitas et al., 2010)[10]. We then extended this verbalizer to Spanish using the adaption of SimpleNLG to Spanish (Soto et al., 2017). We generated Spanish BENGAL datasets and evaluated the aforemen-

[8] Our complete results at https://goo.gl/TBDxCa.

[9] All values are at http://goo.gl/Mg3rE1.

[10] All Portuguese results at http://faturl.com/bengalpt.

Table 2: Excerpt of the features of the datasets used in our evaluation. The datasets B4, B6, B8 and B10 are paraphrased versions of B3, B5, B7 resp. B9 and share similar characteristics.

ID	Name	Doc. $\|C\|$	Tokens $\|T\|$	Entities $\|E\|$	$\|T\|/\|C\|$	$\|E\|/\|C\|$	$\|E\|/\|T\|$
D1	ACE2004	57	21312	306	373.9	5.4	0.01
D2	AIDA/CoNLL-Complete	1393	245008	34929	175.9	25.1	0.14
D8	IITB	104	66531	18308	639.7	176.0	0.28
D11	Microposts2014-Train	2340	40684	3822	17.4	1.6	0.09
D15	OKE 2015 Task 1 evaluation	101	3064	664	30.3	6.6	0.22
B1	BENGAL Path 100	100	1202	362	12.02	3.6	0.30
B2	BENGAL Path Para 100	100	1250	362	12.5	3.6	0.29
B3	BENGAL Star 100	100	3039	880	30.39	8.8	0.29
B5	BENGAL Sym 100	100	2718	725	27.18	7.25	0.26
B9	BENGAL Summary 100	100	2033	637	20.33	6.37	0.31
B11	BENGAL Hybrid 10000	10000	556483	165254	55.6	16.5	0.30
B12	BENGAL Hybrid Long 10	10	9162	2417	241.7	916.2	0.26
B13	BENGAL Star Long 10	10	7369	316	31.6	736.9	0.04

Table 3: Excerpt of micro F1-scores of the annotators for the A2KB experiments on chosen datasets. N/A means that the annotator stopped with an error.

Experiment	Dataset ID	AIDA	Babelfy	Spotlight	Dexter	E.eu	FOX	FRED	FREME	WAT	xLisa-NER	xLisa-NGRAM
A2KB	D1	0.26	0.13	0.18	0.21	0.14	0.13	N/A	0.19	0.25	0.36	0.27
	D2	0.68	0.45	0.54	0.47	0.39	0.51	N/A	0.34	0.67	0.43	0.36
	D8	0.14	0.13	0.26	0.21	0.15	0.10	N/A	0.07	0.14	0.07	0.23
	D11	0.38	0.31	0.45	0.39	0.36	0.32	0.07	0.25	0.40	0.36	0.32
	D15	0.57	0.41	0.46	0.47	0.28	0.55	0.33	0.27	0.53	0.53	0.47
	B1	0.65	0.47	0.69	0.70	0.39	0.50	0.45	0.49	0.61	0.45	0.61
	B2	0.67	0.49	0.68	0.70	0.38	0.54	0.41	0.47	0.61	0.44	0.62
	B3	0.62	0.48	0.57	0.65	0.27	0.47	0.35	0.38	0.53	0.36	0.43
	B5	0.42	0.40	0.42	0.44	0.17	0.34	0.29	0.30	0.35	0.24	0.33
	B9	0.51	0.39	0.57	0.52	0.26	0.43	0.39	0.30	0.46	0.44	0.51
	B11	0.68	0.68	0.69	0.74	0.24	0.49	0.41	0.47	0.65	0.44	0.51
	B12	0.83	N/A	0.79	0.84	0.40	0.73	N/A	0.50	0.79	0.23	0.28
	B13	0.33	0.38	0.33	0.40	0.11	0.17	N/A	0.22	0.45	0.44	0.50

tioned NER and EL systems on them. [11] We also included VoxEL (Rosales-Méndez et al., 2018), a recent gold standard for Spanish. While the extension of BENGAL to Portuguese is an important result in itself, our results also provide additional insights in the NER and EL performance of existing solutions. Our results suggest that existing solutions are mostly biased towards a high precision but often achieve a lower recall on this language. For example, both Spotlight's and Babelfy's recall remain below 0.6 in most cases while their precision goes up to 0.9. This clearly results from the lack of training data for these resource-poor languages. In contrast, the Spanish annotators presented low but consistent results, which confirms the lack of training data of these approaches on Spanish.

5 Discussion and Conclusion

We presented and evaluated BENGAL, an approach for the automatic generation of NER and EL benchmarks. Our results suggest that our approach can generate diverse benchmarks with characteristics similar to those of a large proportion of existing benchmarks in several languages.

Overall, our results suggest that BENGAL benchmarks can ease the development of NER and EL tools (especially for resource-poor languages) by providing developers with insights into their performance at virtually no cost. Hence, BENGAL can improve the push towards better NER and EL frameworks. In future work, we plan to extend the

[11] All Spanish results at http://faturl.com/bengales.

ability of BENGAL to generate longer and more complex sentences as well as the capability of generating different surface forms for a given entity by relying on referring expression models such as NeuralREG model (Castro Ferreira et al., 2018). We also intend to provide thorough evaluations of annotators across other resource-poor languages and create corresponding datasets to push the development of tools to process these languages.

Acknowledgements

This work has been supported by the H2020 project HOBBIT (GA no. 688227) as well as the BMVI projects LIMBO (project no. 19F2029C), OPAL (project no. 19F20284) and also supported by the German Federal Ministry of Education and Research (BMBF) within 'KMU-innovativ: Forschung für die zivile Sicherheit' in particular 'Forschung für die zivile Sicherheit' and the project SOLIDE (no. 13N14456). The authors gratefully acknowledge financial support from the German Federal Ministry of Education and Research within Eurostars, a joint programme of EUREKA and the European Community under the project E! 9367 DIESEL and E! 9725 QAMEL.

References

Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, pages 135–187.

Or Biran and Kathleen McKeown. 2015. Discourse planning with an n-gram model of relations. In *EMNLP*, pages 1973–1977.

Nadjet Bouayad-Agha, Gerard Casamayor, and Leo Wanner. 2014. Natural language generation in the context of the semantic web. *Semantic Web*, 5(6):493–513.

Amparo Elizabeth Cano Basave, Giuseppe Rizzo, Andrea Varga, Matthew Rowe, Milan Stankovic, and Aba-Sah Dadzie. 2014. Making sense of microposts (#microposts2014) named entity extraction & linking challenge. In *Proceedings of 4th Workshop on Making Sense of Microposts*.

Thiago Castro Ferreira, Diego Moussallem, Ákos Kádár, Sander Wubben, and Emiel Krahmer. 2018. Neuralreg: An end-to-end approach to referring expression generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1959–1969. Association for Computational Linguistics.

Philipp Cimiano, Janna Lüker, David Nagel, and Christina Unger. 2013. Exploiting ontology lexica for generating natural language texts from rdf data. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 10–19, Sofia, Bulgaria. Association for Computational Linguistics.

Emilie Colin, Claire Gardent, Yassine Mrabet, Shashi Narayan, and Laura Perez-Beltrachini. 2016. The webnlg challenge: Generating text from dbpedia data. In *Proceedings of the 9th International Natural Language Generation conference*, pages 163–167.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Conference on Empirical Methods in Natural Language Processing-CoNLL*.

Daniel Duma and Ewan Klein. 2013. Generating natural language from linked data: Unsupervised template extraction. In *IWCS*, pages 83–94.

Basil Ell and Andreas Harth. 2014. A language-independent method for the extraction of rdf verbalization templates. In *INLG*, pages 26–34.

Basil Ell, Denny Vrandečić, and Elena Simperl. 2011. Labels in the web of data. *ISWC*.

Marieke van Erp, Pablo Mendes, Heiko Paulheim, Filip Ilievski, Julien Plu, Giuseppe Rizzo, and Joerg Waitelonis. 2016. Evaluating entity linking: An analysis of current benchmark datasets and a roadmap for doing a better job. In *Proceedings of LREC*.

Cláudia Freitas, Paula Carvalho, Hugo Gonçalo Oliveira, Cristina Mota, and Diana Santos. 2010. Second harem: advancing the state of the art of named entity recognition in portuguese. In *quot; In Nicoletta Calzolari; Khalid Choukri; Bente Maegaard; Joseph Mariani; Jan Odijk; Stelios Piperidis; Mike Rosner; Daniel Tapias (ed) Proceedings of the International Conference on Language Resources and Evaluation (LREC 2010)(Valletta 17-23 May de 2010) European Language Resources Association*. European Language Resources Association.

Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. 2016. Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 927–938, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planning. In *Proceedings of ACL*.

Daniel Hewlett, Aditya Kalyanpur, Vladimir Kolovski, and Christian Halaschek-Wiener. 2005. Effective nl paraphrasing of ontologies on the semantic web. In *Workshop on end-user semantic web interaction, 4th int. semantic web conference, galway, ireland*.

Kunal Jha, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2017. All that glitters is not gold–rule-based curation of reference datasets for named entity recognition and entity linking. In *ISWC*.

Pablo N. Mendes, Max Jakob, Andres Garcia-Silva, and Christian Bizer. 2011. DBpedia Spotlight: Shedding Light on the Web of Documents. In *7th International Conference on Semantic Systems (I-Semantics)*, pages 1–8.

David Milne and Ian H Witten. 2008. Learning to link with wikipedia. In *ACM CIKM*.

Rania Mohammed, Laura Perez-Beltrachini, and Claire Gardent. 2016. Category-driven content selection. In *Proceedings of the 9th International Natural Language Generation conference*, pages 94–98.

Andrea Moro, Francesco Cecconi, and Roberto Navigli. 2014. Multilingual word sense disambiguation and entity linking for everybody. In *Proceedings of the 2014 International Conference on Posters & Demonstrations Track-Volume 1272*, pages 25–28. CEUR-WS. org.

Diego Moussallem, Thiago Castro Ferreira, Marcos Zampieri, Maria Claudia Cavalcanti, Geraldo Xexeo, Mariana Neves, and Axel-Cyrille Ngonga Ngomo. 2018. RDF2PT: Generating Brazilian Portuguese Texts from RDF Data. In *LREC*.

Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2017. MAG: A Multilingual, Knowledge-base Agnostic and Deterministic Entity Linking Approach. In *K-CAP: Knowledge Capture Conference*, page 8. ACM.

Yassine Mrabet, Pavlos Vougiouklis, Halil Kilicoglu, Claire Gardent, Dina Demner-Fushman, Jonathon Hare, and Elena Simperl. 2016. Aligning texts and knowledge bases with semantic sentence simplification. *WebNLG 2016*.

Axel-Cyrille Ngonga Ngomo, Lorenz Bühmann, Christina Unger, Jens Lehmann, and Daniel Gerber. 2013. Sorry, i don't speak sparql — translating sparql queries into natural language. In *Proceedings of WWW*, pages 977–988.

Axel-Cyrille Ngonga Ngomo and Frank Schumacher. 2009. Borderflow: A local graph clustering algorithm for natural language processing. In *Computational Linguistics and Intelligent Text Processing*, pages 547–558. Springer.

Andrea Giovanni Nuzzolese, Anna Lisa Gentile, Valentina Presutti, Aldo Gangemi, Darío Garigliotti, and Roberto Navigli. 2015. Open knowledge extraction challenge. In *Semantic Web Evaluation Challenge*.

Sergio Oramas, Luis Espinosa Anke, Mohamed Sordo, Horacio Saggion, and Xavier Serra. 2016. ELMD: an automatically generated entity linking gold standard dataset in the music domain. In *LREC*.

Laura Perez-Beltrachini, Rania Sayed, and Claire Gardent. 2016. Building rdf content for data-to-text generation. In *COLING*, pages 1493–1502.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1375–1384.

Dietrich Rebholz-Schuhmann, Antonio José Jimeno Yepes, Erik M Van Mulligen, Ning Kang, Jan Kors, David Milward, Peter Corbett, Ekaterina Buyko, Elena Beisswanger, and Udo Hahn. 2010. Calbc silver standard corpus. *Journal of bioinformatics and computational biology*.

Michael Röder, Ricardo Usbeck, Daniel Gerber, Sebastian Hellmann, and Andreas Both. 2014. N^3 - A Collection of Datasets for Named Entity Recognition and Disambiguation in the NLP Interchange Format. In *LREC*.

Henry Rosales-Méndez, Aidan Hogan, and Barbara Poblete. 2018. Voxel: A benchmark dataset for multilingual entity linking. In *International Semantic Web Conference*. Springer.

Rolf Schwitter, Marc Tilbrook, et al. 2004. Controlled natural language meets the semantic web. In *Proceedings of the Australasian Language Technology Workshop*, volume 2, pages 55–62.

Amin Sleimi and Claire Gardent. 2016. Generating paraphrases from dbpedia using deep learning. *WebNLG 2016*, page 54.

Alejandro Ramos Soto, Julio Janeiro Gallardo, and Alberto Bugarín Diz. 2017. Adapting simplenlg to spanish. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 144–148.

Kamenka Staykova. 2014. Natural language generation and semantic technologies. *Cybernetics and Information Technologies*, 14(2):3–23.

Nadine Steinmetz, Magnus Knuth, and Harald Sack. 2013. Statistical analyses of named entity disambiguation benchmarks. In *1st Workshop on NLP&DBpedia 2013*, pages 91–102.

Xiantang Sun and Chris Mellish. 2006. Domain independent sentence generation from rdf representations for the semantic web. In *Combined Workshop on Language-Enabled Educational Technology and Development and Evaluation of Robust Spoken Dialogue Systems, European Conference on AI, Riva del Garda, Italy*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, pages 142–147.

George Tsatsaronis, Michael Schroeder, Georgios Paliouras, Yannis Almirantis, Ion Androutsopoulos, Eric Gaussier, Patrick Gallinari, Thierry Artieres, Michael R. Alvers, Matthias Zschunke, and Axel-Cyrille Ngonga Ngomo. 2012. BioASQ: A challenge on large-scale biomedical semantic indexing and question answering. In *AAAI Information Retrieval and Knowledge Discovery in Biomedical Text*.

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 *WWW '15*.

Robert Voyer, Valerie Nygaard, Will Fitzgerald, and Hannah Copperman. 2010. A hybrid model for annotating named entity training corpora. In *Proceedings of the 4th Linguistic Annotation Workshop*.