Event-QA: A Dataset for Event-Centric Question Answering over Knowledge Graphs

Tarcísio Souza Costa, Simon Gottschalk, and Elena Demidova

L3S Research Center, Leibniz Universität Hannover, Hannover, Germany
{souza, gottschalk, demidova}@L3S.de

Abstract. Semantic Question Answering (QA) is the key technology to facilitate intuitive user access to semantic information stored in knowledge graphs. Whereas most of the existing QA systems and datasets focus on entity-centric questions, very little is known about the performance of these systems in the context of events. As new event-centric knowledge graphs emerge, datasets for such questions gain importance. In this paper we present the *Event-QA* dataset for answering event-centric questions over knowledge graphs. *Event-QA* contains 1000 semantic queries and the corresponding English, German and Portuguese verbalisations for EventKG - a recently proposed event-centric knowledge graph with over 970 thousand events.

Resource type: Dataset
Resource DOI: 10.5281/zenodo.3568387
Permanent URL: http://eventcqa.l3s.uni-hannover.de

1 Introduction

Knowledge graphs (KGs) with popular examples including DBpedia [7], Wikidata [18], YAGO [9] and EventKG [2] have recently evolved as an important reference source of semantic information on the Web. Semantic Question Answering (QA) is the key technology to facilitate natural language interfaces to access knowledge graphs. In recent years, a large variety of QA approaches to facilitate effective access to knowledge graphs for end users has been developed [4]. The research and development of QA systems is supported by active development of QA datasets, for example through the QALD (Question Answering over Linked Data) initiative[1].

Existing knowledge graphs are mostly entity-centric, which means they do not sufficiently cover events and temporal relations among entities [2]. As a consequence, existing QA datasets, with recent examples including LC-QuAD [15], LC-QuAD 2.0 [1] and the QALD challenges, mainly focus on entity-centric queries. More recently, event-centric knowledge graphs such as EventKG [2] and knowledge graphs extracted from news (e.g. [6,11]) have been proposed. However,

[1] Question Answering over Linked Data: http://qald.aksw.org

there is a lack of QA datasets dedicated to event-centric questions and only few datasets (e.g. TempQuestions [5] and Saquete et al. [12]) that focus on temporal expressions.

In this paper we introduce the novel dataset *Event-QA* (Event-Centric Question Answering Dataset) for complex event-centric questions over knowledge graphs. With complex we mean that the intended SPARQL queries consist of more than one triple pattern. This corresponds to the definition of complex questions adopted by the LC-QuAD dataset [15]. *Event-QA* includes 1000 semantic queries for the EventKG knowledge graph along with their verbalisations in three languages (English, Portuguese and German), overall resulting in 3000 natural language questions. To the best of our knowledge this is the first QA dataset focused on event-centric questions so far, and the only dataset that targets EventKG. To facilitate easier evaluation of existing QA systems, we also provide a translation of the SPARQL queries in *Event-QA* to DBpedia, where possible.

The aims of *Event-QA* are to: 1) Provide a representative selection of relations involving events in the knowledge graph, so as to ensure the diversity of the resulting event-centric semantic queries. To achieve this goal, we approach the query generation automatically via a random walk through the knowledge graph, starting from randomly selected relations. 2) Ensure the quality of the natural language questions. To this end, the resulting queries are manually translated into natural language expressions in English, Portuguese and German. These translations are verified manually to ensure their quality.

The main contributions are as follows: (i) The *Event-QA* dataset containing 3000 event-centric natural language questions (i.e. 1000 in each language) with the corresponding SPARQL interpretations for the EventKG knowledge graph. These questions exhibit a high variety of SPARQL queries and their verbalisations. (ii) An approach for an automatic generation of semantic queries for an event-centric QA dataset from a knowledge graph. (iii) An open-source extensible framework for automatic dataset generation to facilitate dataset maintenance and updates.

2 Relevance

Relevance to the Semantic Web community and society: Question Answering (QA) [4] is the key technology to provide end users access to structured semantic data in an intuitive way. Research on the development of QA applications is of interest for several scientific communities including Semantic Web, NLP and HCI [13]. Semantic QA approaches automatically translate user queries posed in a natural language into structured queries, e.g. into the SPARQL query language. Whereas current research is mostly focused on entity-centric questions, events are still underrepresented in semantic reference sources and their corresponding QA resources, including datasets and benchmarks.

Relevance for Question Answering applications: Event-centric reference sources are still very rare, with the first event-centric knowledge graphs such as EventKG being introduced only recently. Often, event-centric information is

spread across entity-centric knowledge graphs, is less annotated and more complex in comparison to the entity-centric information, and is more difficult to retrieve. As a consequence, existing QA datasets such as LC-QuAD [15] and QALD are mainly entity-centric. Specialised event-centric QA datasets are currently non-existing. The provision of such resource can bring novel perspective in the Question Answering research and facilitate further development of QA approaches in the context of events.

Impact in supporting the adoption of Semantic Web technologies: Event-centric information is of crucial importance for researchers and practitioners in a variety of domains, including journalists, media analysts and researchers in Digital Humanities. Current and historical events of global importance such as the Brexit, the Olympic Games and the US withdrawal from the nuclear arms treaty with Russia as well as representations of these events across different sources are the subject of current research activities in several fields including Digital Humanities and Web Science (see e.g. [3,10]). Event-centric repositories and the corresponding QA systems can help to answer relevant questions and support these studies. Overall, the provision of intuitive access methods to the semantic reference resources can facilitate a wider adoption of semantic technologies by researchers and practitioners in these fields.

3 Problem Statement

Semantic Question Answering (QA) is a process of translating user questions expressed in a natural language into the corresponding semantic queries for a given knowledge graph.

The goal of this work is to create a Question Answering dataset to support the development and evaluation of QA approaches for event-centric questions. In the context of this work, events are real-world happenings of societal importance, typically found in encyclopedic sources such as Wikipedia and the corresponding knowledge graphs like EventKG [2], DBpedia [7] and Wikidata [18]. Examples of relevant events include military conflicts, sports tournaments and political elections.

The *Event-QA* dataset, referred to as β in the following, consists of a set of semantic queries. Each semantic query $q \in \beta$ mentions at least one event. Furthermore, each query q is aligned with one or more verbalisations q_{NL}, i.e. questions expressed in natural language(s).

In this section, we first define the notion of the knowledge graph. Then, we discuss requirements for an event-centric QA dataset.

Definition 1. ***Knowledge Graph.*** *A knowledge graph KG is a labelled multi-graph $KG = (V, R_v, R_l, L)$. $V = E_v \cup E_n$ is a set of nodes in KG. The set E_v represents real-world events. The set E_n represents real-world entities. L is a set of literals. R_v and R_l are sets of edges. An edge in R_v connects two nodes in V and an edge in R_l connects a node in V with a literal in L.*

Literals represent specific properties of events and entities in a knowledge graph, e.g. the start time of an event or the entity name.

Definition 2. ***Relation.*** *$Rel \subseteq V \times R_v \times V$ denotes the set of real-world relations between particular instances of events and entities in a knowledge graph KG.*

A *query graph* is a subgraph of the knowledge graph. A query graph includes a subset of nodes and edges of the knowledge graph, as well as a set of variables representing such nodes and edges. As the focus of this work is on event-centric questions, at least one node in the query graph represents an event.

Definition 3. ***Query graph.*** *A query graph $q = (V', R'_v, R'_l, L', U)$ is a subgraph of the knowledge graph $KG = (V, R_v, R_l, L)$, $V = E_v \cup E_n$, where $V' \subset V$, $R'_v \subset R_v$, $R'_l \subset R_l$, $L' \subset L$, and U is a set of variables. Each variable $u \in U$ maps to a node of KG. At least one node in the query graph represents an event: $\exists v' \in E_v : v' \in V' \vee \exists u' \in U : u' \mapsto v'$.*

A semantic query $q \in \beta$ in the dataset includes a query graph, a query type and optionally a set of constraints. The query type represents a projection operator such as SELECT and ASK or an aggregation operator such as COUNT. Constraints can e.g. be used to restrict the time period of interest for the query.

Definition 4. ***Semantic query.*** *A semantic query consists of: 1) a query graph, 2) a query type, 3) an optional set of constraints.*

A semantic query can be expressed in the SPARQL query language.

In order to facilitate an effective assessment of performance of QA systems for event-centric queries, the dataset β has to include QA tasks of sufficient difficulty. In particular this means that 1) the mapping between the natural language question and the semantic query is non-trivial, and 2) the queries in β exhibit a variety of patterns.

To achieve this goal, the dataset should satisfy conditions regarding:

- the complexity of semantic queries, i.e. the queries should include more than one triple pattern;
- the diversity of the semantic queries, i.e. the semantic queries in the dataset should be dissimilar to each other; and
- the diversity of the query verbalisations, i.e. the verbalisations of the queries in the dataset should be dissimilar to each other.

4 Dataset Generation Approach

In this section we present our approach to generate an event-centric QA dataset containing complex and diverse queries given a knowledge graph. This approach can be adopted for any knowledge graphs that contains a set of identifiable named events.

4.1 Semantic Query Generation Pipeline

In order to generate a set of complex and diverse queries that are meaningful to human users we adopted a two-stage approach. First, we employed an initial

version of the semantic query generation pipeline presented later in this section to automatically generate an initial sample of semantic queries. Using these queries, we conducted the first annotation stage, where we manually created verbalisations of the suggested semantic queries and marked the queries that did not appear meaningful. We analysed the patterns in the annotations and used the collected observations to fine-tune the query generation pipeline. Finally, we used the pipeline to generate the final set of queries. Subsequently, we manually annotated this set to obtain the query verbalisations in the *Event-QA* dataset.

Our semantic query generation pipeline for generating a single query is illustrated in Fig. 1.

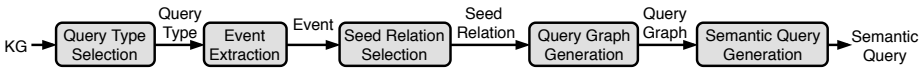

Fig. 1. Overview of the *Event-QA* pipeline. Given a knowledge graph as input, one execution of this pipeline leads to the generation of one semantic query.

For each query to be generated, the *Event-QA* pipeline includes the following steps:

1. **Query Type Selection:** A query type is selected randomly (i.e. ASK, SELECT or COUNT).
2. **Event Extraction:** A named event node from the knowledge graph is randomly selected together with all relations connected to that node.
3. **Seed Relation Selection:** A relation is randomly selected from the list of all relations involving the previously selected event. We refer to such relation as seed relation. The seed relation includes at least one event, as ensured by step 2.
4. **Query Graph Generation:** A query graph is generated as a sub-graph of the knowledge graph augmented with variables.
   (a) **Sub-Graph Generation:** To generate a sub-graph containing more than one relation, we conduct a random walk over the knowledge graph starting from the seed relation.
   (b) **Augmentation with Variables:** The sub-graph is complemented with variables to obtain a query graph.
5. **Semantic Query Generation:** The query graph is augmented with the query type and optionally temporal constraints to build the semantic query. The resulting semantic query is translated into the SPARQL query language.
6. **Query Verbalisation:** For each SPARQL query resulting from the *Event-QA* pipeline, the corresponding verbalisation is defined manually.

An example seed relation is shown in Fig. 2. Based on this seed relation, the query graph shown in Fig. 3 can be created with the random walk based approach. Finally, the query graph is translated into a SPARQL query for a knowledge graph (Fig. 4).

4.2 Query Type Selection

The pipeline starts with a random selection of the type for the semantic query to be generated. The query types included in the *Event-QA* dataset are ASK, SELECT and COUNT. These types correspond to the SPARQL query forms[2]. ASK queries determine whether a query pattern has a solution. SELECT returns variables and their bindings. COUNT computes the number of results for a given expression.

4.3 Event Extraction

In order to ensure that all queries in the dataset include at least one event, we start the query graph generation process by randomly picking one event from the event set E_v. This event and its associated relations build the input for the next step.

4.4 Seed Relation Selection

We randomly choose a relation $(n_1, r, n_2) \in Rel$ that belongs to the set of relations referring to the event picked in the previous step. This relation takes the role of the *seed relation* in the current execution of the *Event-QA* pipeline. As our goal is to generate complex event-centric queries, a seed relation needs to fulfil certain criteria: (i) the seed relation needs to include at least one event ($n_1 \in E_v \vee n_2 \in E_v$) which is guaranteed by step 2 in our pipeline, and (ii) at least one of the nodes included in the relation needs to be part of another relation in the knowledge graph. Fig. 2 provides an example of a seed relation (`dbr:2002_German_Grand_Prix`, `dbo:fastestDriverTeam`, `dbr:Scuderia_Ferrari`)[3].

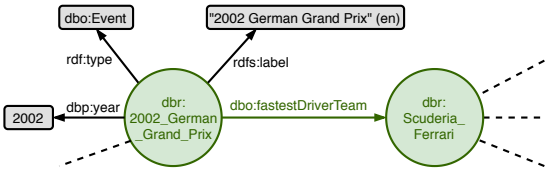

Fig. 2. Seed relation example. Given the event labelled "2002 German Grand Prix" as the starting point, the relation (`dbr:2002_German_Grand_Prix`, `dbo:fastestDriverTeam`, `dbr:Scuderia_Ferrari`) (marked in green) was randomly selected from the knowledge graph as a seed relation.

[2] `https://www.w3.org/TR/sparql11-query`
[3] `dbr` is the prefix of the DBpedia resource identifier: http://dbpedia.org/resource/.

4.5 Query Graph Generation

Based on the seed relation extracted in the previous step, a query graph is extracted. This query graph is a sub-graph of the knowledge graph that contains the seed relation and variables. Thus, two steps are required to create the query graph: (i) sub-graph generation and (ii) augmentation with variables.

Sub-Graph Generation We obtain a sub-graph of the knowledge graph through a random walk procedure. That way, we aim to ensure the diversity of the generated semantic queries.

The seed relation constitutes the initial sub-graph. Through a random walk we incrementally extend the sub-graph by adding new relations to the nodes already included in the sub-graph in a way that the sub-graph remains connected. In each step, we randomly select a node of the sub-graph. Then, we randomly select an edge from the knowledge graph involving that node. If this edge is not part of the sub-graph yet, the corresponding relation is added to the sub-graph.

Fig. 3 is an example of a query graph from the seed relation in Fig. 2. Another relation was added (`?event`, `dbo:secondTeam`, `dbr:Williams_Grand_Prix_Engineering`) with the variable `?event` defined that can be mapped to `dbr:2002_German_Grand_Prix`. Additionally, the event labelled "2002 German Grand Prix" was replaced with a variable (we describe the methods for selecting such variables in more detail later). The resulting query graph can be represented in natural language as: *In which competition in 2002 did Ferrari appear as the fastest driver team and Williams as the second team?*

The random walk continues until the termination condition is met. In particular, we apply a threshold to restrict the maximal number of relations of the query. The value of this threshold is decided based on the manual annotations in the fine-tuning stage. Here we observed that the majority of the queries that included three or more relations were difficult to interpret for humans; such queries typically include relations that do not posses intuitive real-world interpretations; therefore in the current version of the *Event-QA* dataset, we restrict the threshold value to two[4]. In principle, the proposed approach is flexible with respect to the threshold value such that the threshold can be increased to enable more than two relations in a semantic query in cases where such configurations are meaningful, e.g. in domain-specific knowledge graphs.

We illustrate the issue of too high complexity at the example from Fig. 3. When the threshold value was above two, a third relation can be added to the query graph, e.g. `?event dbo:secondDriver dbr:Juan_Pablo_Montoya`. The corresponding English verbalisation of this query could be as follows: *In which competition in 2002, where Juan Pablo Montoya was the second driver, did Ferrari appear as the fastest driver team and Williams as the second team?* This query was annotated as too complex and unnatural during the first annotation round.

[4] Note that this limit does not include temporal constraints, meaning that it is possible for a query to have two relations and additional temporal constraints.

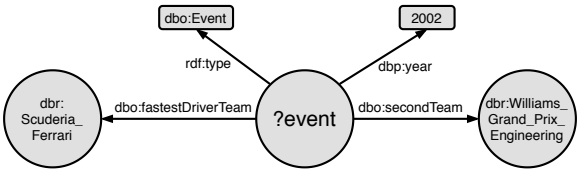

Fig. 3. Query graph example. Following the random walk, another edge was added to the initial sub-graph shown in Fig. 2. The event node was replaced with a variable.

Augmentation with Variables In this step we systematically define the variable $u \in U$ that is selected to build the query graph, given the sub-graph resulting from the previous step. A variable can replace either a literal or a node in the query graph. We assign variables randomly. An example of a resulting query graph is provided in Fig. 3.

Whereas we experimented with multiple variables per query graph and a completely random assignment procedure, our observations of the annotations during the pipeline tuning resulted in constraints for the variable selection. These constraints increase the likelihood that the resulting queries can lead to a meaningful verbalisation.

In particular, in the final pipeline configuration, we:

- add at most one variable to the query graph;
- do not include variables to the ASK queries;
- avoid variables as leaf nodes of the query graph, where possible;
- avoid variables representing information redundant in the query graph. For example, time information is often represented as part of the event name (e.g. "*2002 German Grand Prix*"), as well as event start and end time.
- avoid variables representing literals containing time-related information in the case of queries of type COUNT.

4.6 Semantic Query Generation

Together, the query type selected in Section 4.2, the query graph generated in Section 4.5 and optional temporal constraints described in the following constitute a semantic query. This semantic query is translated into a SPARQL query for a specific knowledge graph automatically.

Augmentation with Temporal Constraints When possible, temporal constraints can be included in the semantic query. Such temporal information could for example denote the validity time of a relation or an existence time of an entity included in the query graph. Based on this information, we define a time interval of interest. Then, we randomly select one of the following temporal constrains and add it to the query: (i) within the time interval, (ii) after the time interval, or (iii) before the time interval.

Fig. 4 provides an example of a SPARQL query, given the query graph in Fig. 3, with an added temporal constraint.

```
SELECT (COUNT(DISTINCT(?event) AS ?count)) WHERE {
  ?event rdf:type dbo:Event .

  ?event dbo:fastestDriverTeam dbr:Scuderia_Ferrari .
  ?event dbo:secondTeam dbr:Williams_Grand_Prix_Engineering .

  ?event dbp:year ?year .
  FILTER ( ?year > "2001"^^xsd:integer)
}
```

Fig. 4. SPARQL Query example for the query graph in Fig. 3 in DBpedia.

4.7 Pipeline Fine-tuning with Manual Annotations

The pipeline fine-tuning and the creation of query verbalisations (i.e. natural language representations of semantic queries) in the *Event-QA* dataset are conducted manually. In addition to the authors of this work, three post-graduates with expertise in SPARQL and RDF participated in the verbalisation. To facilitate these tasks, we implemented a Web interface that displayed the SPARQL queries generated by the pipeline shown in Fig. 1.

In order to collect input for the pipeline fine-tuning, we used the interface to observe any systematic patterns that make the queries difficult to understand or difficult to translate into meaningful natural language questions. Our interface provided the following instructions for annotation of a given SPARQL query:

1. Read the SPARQL query and think of the question it represents.
   - If you do not understand the query, select the "I do not understand the query" option. Leave a comment on what makes it difficult to understand the query. Click "continue". The next query will be shown.
2. Do you think that a human user would ask the question represented by this query?
   - If you think that is a question a user would not ask, please select the option "A user would not ask this question". Leave a comment to explain why. Click "continue". The next query will be shown.

From this annotation process on a sample of queries, we have gained insights into the query complexity and the allocation of variables.

Query Complexity: As stated in Section 4.5, we observed that queries which included more than two relations were less meaningful and more difficult to understand for the participants. As a result, we restrict the complexity of the queries in the dataset to a maximum of two relations.

Allocation of Variables: An appropriate allocation of variables is one of the most critical issues to generate queries understandable and meaningful for the participants. In particular, queries containing more than one variable are often difficult to understand or do not have any meaningful natural language representation. The same observation applies to the ASK queries that contain

variables. Finally, queries that include multiple relations and contain variables at the leaf nodes of the query graph do not result in a meaningful natural language interpretation; rather the variables should be allocated at the nodes that connect several relations in the query graph. For the creation of the final set of queries, we introduced the corresponding rules in the variable augmentation step of the semantic query generation pipeline.

4.8 Query Verbalisation

The SPARQL queries generated using the fine-tuned semantic query generation pipeline are annotated with English, Portuguese and German questions. For the English questions, each SPARQL query verbalisation was manually confirmed as in the first annotation step described in Section 4.7. For the annotations, we formulated the following instructions:

- Try to formulate the question in a way that sounds natural.
- If possible, vary the language expressions you use for different queries.

Finally, native Portuguese and German speakers among the authors provided high quality translations of the English queries in the corresponding language.

Note that although approaches to automatic generation of NL expressions from SPARQL queries such as SPARQL2NL [8] exist, and have been tested by us as we developed the dataset, we observed two main problems: 1) Typically, automatically generated NL-expressions do not result in intuitive sentences for complex SPARQL queries including more than one relation; and 2) artificial NL-expressions used as a suggestion for manual reformulation do not help to speed up the manual translation process, as the user has to understand a complex and potentially erroneous NL-expression in addition to the original SPARQL query. Therefore, to ensure the NL-quality and the efficiency of the manual annotation process, we stick to the manual query verbalisation. We believe that although this process can be automated in principle, this would require significant further development of the automatic translation methods.

5 Application to EventKG

The knowledge graph adopted for the creation of queries in *Event-QA* is EventKG [2] – a multilingual large-scale temporal knowledge graph. EventKG V2.0, released in 03/2019 contains over 970k contemporary and historical events and over 2.8 million temporal relations extracted from DBpedia, Wikidata and YAGO and several semi-structured sources.

5.1 EventKG as a Knowledge Graph

EventKG can be directly expressed as a knowledge graph according to Definition 1 in Section 3, using EventKG's ontology that is based on the Simple Event Model (`sem`) [17]. The set of events E_v consists of all EventKG resources marked as `sem:Event`, the set of entities E_n consists of all EventKG resources

marked as an instance of sem:Core, but not as sem:Event. The set of relations R_v corresponds to EventKG's eventKG-s:Relation instances[5] plus any other relations using predefined properties that are part of the EventKG schema such as dbo:nextEvent and sem:hasPlace. In EventKG, begin and end times of events and entities are represented by sem:hasBeginTimeStamp and sem:hasEndTimeStamp properties. Thus, these relations make up the set of relations R_l.

5.2 Applying the *Event-QA* Query Generation Algorithm to EventKG

Fig. 5 illustrates an example of how the *Event-QA* query graph generation approach is applied to the knowledge graph based on EventKG. eventKG-r-01 takes the role of the seed relation in this example. This relation connects an event node (e.g. dbr:1973_Uruguayan_Primera_División) with the entity dbr:Peñarol.

In this example, the first application of the random walk leads to the selection of the event node which is connected to the relation node eventKG-r-02. In the next iteration, the relation node eventKG-r-02 is added to the sub-graph, which connects the sub-graph to the entity dbr:Uruguay.

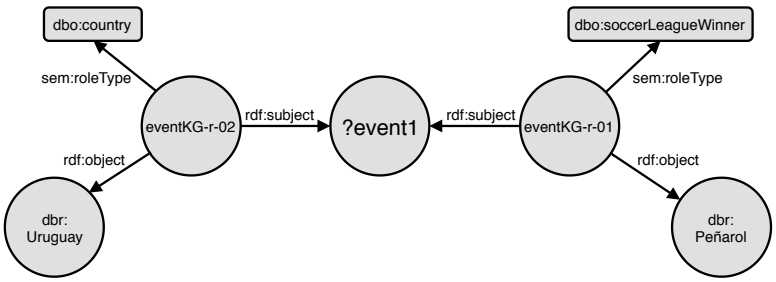

Fig. 5. An example of a query graph created when applying the *Event-QA* algorithm to EventKG. For readability reasons, we denote the nodes using their DBpedia resource identifiers (dbr).

To obtain the query graph, a randomly selected node of the sub-graph is replaced with a variable. In the example illustrated, the event node plays the role of a variable which results in the SPARQL query shown in Fig. 6. This query can be verbalised as: *"In which soccer leagues did Peñarol win in Uruguay?"*.

5.3 Translation of EventKG Queries to DBpedia

As *Event-QA* is a dataset with event-centric queries, it is expected to perform best on an event-centric knowledge graph such as EventKG. However, to facilitate an easier evaluation of existing QA systems using the *Event-QA dataset*,

[5] In EventKG, most relations are modelled as resources that point to the relation's subject (via rdf:subject), object (rdf:object) and property (sem:roleType).

```
SELECT DISTINCT ?event WHERE {
  ?relation1 rdf:object ?entity1 .
  ?relation1 rdf:subject ?event .
  ?relation1 sem:roleType dbo:country .

  ?relation2 rdf:object ?entity2 .
  ?relation2 rdf:subject ?event .
  ?relation2 sem:roleType dbo:soccerLeagueWinner .

  ?entity1 owl:sameAs dbr:Uruguay .
  ?entity2 owl:sameAs dbr:Peñarol .
}
```

Fig. 6. Translation of the query graph in Fig. 5 into a SPARQL query for EventKG.

we also provide a translation of the SPARQL queries to the English DBpedia knowledge graph due to its popularity in the existing QA systems, where possible. This translation is performed by transforming the `eventKG-s:Relation` instances to the DBpedia triples using the `sem:roleType` values. Whenever possible, start and end dates from EventKG are mapped to the temporal DBpedia predicates (e.g. `dbp:year`, `dbo:date` and `dbo:startDate`).

Using this approach, 307 of the SPARQL queries that target EventKG in *Event-QA* can be translated to the English DBpedia. This number can be explained by the underlying structural and content differences between these two knowledge graphs, whereas DBpedia is not specialised on events and covers much less event-centric information compared to EventKG. The differences are as follows: i) Semantic queries generated using EventKG contain relations that originate from a variety of sources (e.g. Wikidata and non-English DBpedia versions); these relations are not always present in the English DBpedia. (ii) Event instances are underrepresented in the English DBpedia, compared to EventKG. (iii) DBpedia does not contain unified dedicated temporal properties, so that the automatic mapping of temporal properties for the individual entities and events fails in many cases.

6 Evaluation and Dataset Characteristics

The quality of queries and the correctness of the translation between SPARQL and natural language is essential towards the quality of a Question Answering dataset. Therefore, we describe our methods that ensure these criteria and provide statistics and example queries. Additionally, we provide insights into the diversity and complexity of the resulting *Event-QA* dataset.

6.1 Dataset Quality

In order to access the correctness of the translation between the SPARQL queries and the corresponding natural language expressions, we manually verified all

the queries included in the dataset and corrected verbalisations in case any issues were observed. This way we ensure that the SPARQL queries and their verbalisations possess equivalent semantics.

6.2 Dataset Statistics and Examples

Event-QA consists of 1000 verified semantic queries. Fig. 7 lists six example query verbalisations that are part of the dataset.

When did the Excitante music festival finish in Argentina? ↪ PT: Quando o festival de música Excitante terminou na Argentina? ↪ DE: Wann ging das argentinische Musical Excitante zu ende?
Give me a list of football events won by Dynamo Kyiv.
How many events chaired by Derek Shaw were part of the EFL championships?
Was Whiplash a film directed by Damien Chazelle and the opening film of the 2014 Sundance Film Festival?

Fig. 7. Example natural language queries in *Event-QA*. For the first example query, we also show the Portuguese and the German natural language translation.

In total, 1005 different events, 1655 different entities and 309 different relation labels occur within the 1000 SPARQL queries. The most frequent relation labels are `dbo:commander`, `dbo:award`, `dbo:city`, `dbo:battle`, `dbo:birthPlace`, `dbo:sport` and `dbo:team`.

6.3 Complexity and Diversity

In order to assess to which extent the dataset generated in this work satisfies the conditions specified in Section 3, we define metrics to assess the complexity and the diversity of semantic queries as well as the diversity of their verbalisations. For comparison, we compute the values of such metrics for *Event-QA*, as well as for the QA datasets LC-QuAD [15], Saquete et al. [12] and TempQuestions [5]. The results are shown in Table 1 and are now explained in more detail. Overall, the datasets considered behave similar with respect to the complexity and diversity metrics, whereas only *Event-QA* focuses on events.

Query Complexity The complexity of a semantic query grows with an increasing number of relations. For simplicity, we assume linear growth. Based on this assumption, we measure the complexity of a semantic query as the number of relations included in the query graph. The overall complexity of the dataset is computed as an average complexity of the queries it contains. Table 1 shows that *Event-QA* queries have 2 relations on average and thus are as complex as the LC-QuAD queries. As discussed in Section 4.7, queries of higher complexity do not appear to be meaningful in terms of user questions.

Table 1. Complexity, query diversity and verbalisation diversity (in English, Portuguese and German) of *Event-QA* in comparison to other QA datasets. Complexity and query diversity are only reported if SPARQL queries are available.

Dataset	Complexity	Query Diversity	Verbalisation Diversity
LC-QuAD	2.0	0.95	0.87
Event-QA	2.0	**0.98**	0.82 (EN), 0.86 (PT), 0.87 (DE)
Saquete et al. [12]	-	-	0.84
TempQuestions [5]	-	-	**0.89**

Query Diversity The dataset β is semantically diverse if its semantic queries are dissimilar. Similar queries share events, entities and relations. Therefore, we assess the similarity of semantic queries using the Jaccard coefficient applied to the set of nodes and edges contained in the corresponding query graphs. To access the diversity of the queries in the dataset as a whole, we compute the diversity as an average dissimilarity value across all query pairs:

$$diversity(\beta) = 1 - \frac{\sum_{(q_i,q_j)\in\beta} similarity(q_i, q_j)}{\binom{|\beta|}{2}}, \tag{1}$$

where (q_i, q_j) is a query pair in the dataset with $q_i \neq q_j$ and $|\beta|$ is the number of queries in the dataset.

Semantic queries in *Event-QA* are slightly more diverse than those from LC-QuAD, as per Table 1. In difference to LC-QuAD, our approach does not depend on templates to build semantic queries. Instead, query graphs in *Event-QA* are build by randomly taken decisions (i.e. random seed selection and random walk).

Verbalisation Diversity The dataset β is textually diverse if the verbalisations q_{NL} of its queries are dissimilar. Similar verbalisations share terms. Therefore, we assess the similarity of the verbalisations using the cosine similarity of the verbalisations represented as tf-idf term vectors. To access the diversity of query verbalisations in the dataset as a whole, we compute the diversity as an average dissimilarity value across all pairs of verbalisations:

$$diversity(\beta_{NL}) = 1 - \frac{\sum_{(q_{NL_i},q_{NL_j})\in\beta} cos(q_{NL_i}, q_{NL_j})}{\binom{|\beta|}{2}}, \tag{2}$$

where (q_{NL_i}, q_{NL_j}) is a verbalisation pair in the dataset and $|\beta|$ is the number of queries in the dataset.

Table 1 reveals that the verbalisations of TempQuestions are most diverse, which was expected, given that this dataset actually combines three different source datasets. *Event-QA*'s English and Portuguese query verbalisations are more diverse than in Saquete et al. and only slightly less diverse than in LC-QuAD, even though the focus of our dataset on event-centric queries gives high impact for words typically associated with events such as "event", "season" and "battle".

7 Availability & Sustainability

The *Event-QA* homepage[6] provides a description of the dataset and how to use and cite the resource. In addition, we provide permanent access to the data on Zenodo[7] released under CCBY 4.0[8]. The DOI of the dataset is: **10.5281/zenodo.3568387**. The data we provide includes:

- **Event-QA**: Event-Centric Question Answering dataset in the QALD JSON format [16] for EventKG. For each query we provide a SPARQL expression for EventKG (and where possible for DBpedia), verbalisations in English, Portuguese and German, and gold-standard answers.
- **Predicates**: List of DBpedia predicates used in the dataset.
- **VoID description**: Machine readable description of the dataset in RDF.
- List of DBpedia **events** and **entities** covered by the dataset.

The dataset generation pipeline is available as open source software on GitHub[9] under the MIT License[10].

Regarding sustainability, we plan to provide regular updates to ensure compatibility with future versions of EventKG. *Event-QA* will be integrated into the FAIR benchmarking platform GERBIL QA[11], which powered past QALD challenges.

8 Related Work

Existing QA datasets greatly vary in size, content (question/answer pairs vs. semantic queries), complexity (i.e. the number of triple patterns in the semantic queries), coverage of natural languages, quality of natural language expressions and target knowledge bases. Even more importantly, existing QA datasets for complex questions such as LcQuAD [15] do not sufficiently address questions regarding events. This can be partially attributed to the fact that most of the popular KGs are entity-centric and event-centric KGs such as EventKG were developed only recently. Although few QA datasets (TempQuestions [5] and Saquete et al. [12]) focus on temporal expressions, they do not sufficiently cover event-centric questions. Another important shortcoming of many existing datasets is the deficiency in natural language expressions quality, in particular in cases where such expressions are generated through automated processes, crowd-sourcing with non-expert users (e.g. LCQuAD 2.0 [1], ComplexWebQuestions [14]) or missing manual verification in general ([16]).

Event QA provides complex event-centric questions in three languages, while adopting automatic question generation process along with manual verbalisation by expert users to ensure quality. With its focus on event-centric questions, and manual verification of multilingual natural language expressions, Event-QA takes a unique position among the available QA datasets.

[6] http://eventcqa.l3s.uni-hannover.de/
[7] https://doi.org/10.5281/zenodo.3568387
[8] https://creativecommons.org/licenses/by/4.0/
[9] https://github.com/tarcisiosouza/Event-QA
[10] https://opensource.org/licenses/MIT
[11] http://gerbil-qa.aksw.org/gerbil/

9 Conclusion

In this paper we presented *Event-QA* – an event-centric dataset for semantic Question Answering. This dataset is generated through a random walk-based approach applied to the EventKG knowledge graph to ensure diversity of the resulting queries. The translation of the resulting semantic queries into natural language is performed and verified manually to ensure the quality of the verbalisation. The resulting dataset contains 1000 complex event-centric questions that can be used for benchmarking question answering systems. The query verbalisations in the *Event-QA* are available in English, Portuguese and German.

References

1. M. Dubey and et al. LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia. In *ISWC 2019*, Cham, 2019.
2. S. Gottschalk and E. Demidova. EventKG: A Multilingual Event-Centric Temporal Knowledge Graph. In *Proc. of the ESWC 2018*. Springer, 2018.
3. S. Gottschalk et al. Towards Better Understanding Researcher Strategies in Cross-Lingual Event Analytics. In *TPDL*, 2018.
4. K. Höffner et al. Survey on Challenges of Question Answering in the Semantic Web. *Semantic Web*, 8(6):895–920, 2017.
5. Z. Jia, A. Abujabal, R. Saha Roy, J. Strötgen, and G. Weikum. TempQuestions: A Benchmark for Temporal Question Answering. In *The Web Conference*, 2018.
6. K. Leetaru and P. A. Schrodt. GDELT: Global Data on Events, Location, and Tone, 1979-2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer, 2013.
7. J. Lehmann et al. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web*, 6(2), 2015.
8. A.-C. N. Ngomo et al. Sorry, I don't Speak SPARQL: Translating SPARQL Queries into Natural Language". In *WWW*, 2013.
9. T. Rebele et al. YAGO: A Multilingual Knowledge Base from Wikipedia, Wordnet, and Geonames. In *Proc. of the ISWC'16*, 2016.
10. R. Rogers. *Digital Methods*. MIT Press, 2013.
11. M. Rospocher et al. Building Event-Centric Knowledge Graphs from News. *J. Web Sem.*, 37-38:132–151, 2016.
12. E. Saquete et al. Enhancing QA Systems with Complex Temporal Question Processing Capabilities. *J. Artif. Intell. Res.*, 35:775–811, 2009.
13. S. Shekarpour et al. Question Answering on Linked Data: Challenges and Future Directions. In *Companion Proc. of the WWW'16*, 2016.
14. A. Talmor and J. Berant. The web as a knowledge-base for answering complex questions. *CoRR*, abs/1803.06643, 2018.
15. P. Trivedi et al. LC-QuAD: A Corpus for Complex Question Answering over Knowledge Graphs. In *Proc. of the ISWC'17*, 2017.
16. R. Usbeck et al. 9th Challenge on Question Answering over Linked Data (QALD-9) (invited paper). In *(QALD-9)*, pages 58–64, 2018.
17. W. R. Van Hage, V. Malaisé, R. Segers, L. Hollink, and G. Schreiber. Design and Use of the Simple Event Model (SEM). *Web Semantics*, 9(2):128–136, 2011.
18. D. Vrandečić. Wikidata: A New Platform for Collaborative Data Collection. In *Proc. of the WWW '12 Companion*, pages 1063–1064. ACM, 2012.