LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia

Mohnish Dubey[1,2], Debayan Banerjee[1,2], Abdelrahman Abdelkawi[2,3], and Jens Lehmann[1,2]

[1] Smart Data Analytics Group (SDA), University of Bonn, Germany
{ dubey, jens.lehmann}@cs.uni-bonn.de , debayan@uni-bonn.de
[2] Fraunhofer IAIS, Bonn, Germany
{mohnish.dubey, jens.lehmann}@iais.fraunhofer.de
[3] RWTH Aachen, Germany

Resource Type: Dataset
Website and documentation: `http://lc-quad.sda.tech/`
Permanent URL: `https://figshare.com/projects/LCQuAD_2_0/62270`

Abstract. Providing machines with the capability of exploring knowledge graphs and answering natural language questions has been an active area of research over the past decade. In this direction translating natural language questions to formal queries has been one of the key approaches. To advance the research area, several datasets like WebQuestions, QALD and LCQuAD have been published in the past. The biggest data set available for complex questions (LCQuAD) over knowledge graphs contains five thousand questions. We now provide LC-QuAD 2.0 (Large-Scale Complex Question Answering Dataset) with 30,000 questions, their paraphrases and their corresponding SPARQL queries. LC-QuAD 2.0 is compatible with both Wikidata and DBpedia 2018 knowledge graphs. In this article, we explain how the dataset was created and the variety of questions available with examples. We further provide a statistical analysis of the dataset.

1 Introduction

In the past decade knowledge graphs such as DBpedia[8] and Wikidata[14] have emerged as major successes by storing facts in linked data architecture. DBpedia recently decided to incorporate the manually curated knowledge base of Wikidata [7] into its own knowledge graph[4]. Retrieving factual information from these knowledge graphs has been a focal point of research. Question Answering over Knowledge graphs(KGQA) is one of the techniques used to achieve this goal. In KGQA, the focus is generally on translating a natural language question to a formal language query. This task has generally been achieved by rule-based systems [6]. However, in the last few years, more systems using machine learning for this task have evolved. QA Systems have achieved impressive results working on simple questions [9] where a system only looks at a single fact consisting of a <subject - predicate - object> triple. On the other hand, for Complex questions (which require retrieval of answers based on more than one triple) there is still ample scope for improvement.

[4] we refer this as 'DBpedia2018' further in this article.

Datasets play an important role in AI research as they motivate the evolution of the current state of the art and the application of machine learning techniques that benefit from large-scale training data. In the area of KGQA, datasets such as WebQuestions, SimpleQuestions and the QALD challenge datasets have been the flag bearers. LCQuAD version 1.0 was an important breakthrough as it was the largest complex question dataset using SPARQL queries at the time of its release. In this work, we present LC-QuAD 2.0 (Large-Scale Complex Question Answering Dataset 2.0) consisting of 30,000 questions with paraphrases and corresponding SPARQL queries required to answer questions over Wikidata and DBpedia2018. This dataset covers several new question type variations compared to the previous release of the dataset or to any other existing KGQA dataset (see comparison in Table 1). Apart from variations in the type of questions, we also paraphrase each question, which allows KGQA machine learning models to escape over-fitting to a particular syntax of questions. This is also the first dataset that utilises qualifier [5] information for a fact in Wikidata, which allows a user to seek more detailed answers (as discussed in Section 4).

The following are key contributions of this work:

- Provision of the largest dataset of 30,000 complex questions with corresponding SPARQL queries for Wikidata and DBpedia 2018.
- All questions in LCQuAD 2.0 also consist of paraphrased versions via crowdsourcing tasks. This provide more natural language variations for the question answering system to learn from and avoid over-fitting on a small set of syntactic variations.
- Questions in this dataset have a good variety and complexity levels such as multi-fact questions, temporal questions and questions that utilise qualifier information.
- This is the first KGQA dataset which contains questions with dual user intents and questions that require SPARQL string operations (Section 4.2).

This article is organised into the following sections: (2) Relevance and significance (3) Dataset Creation Workflow (4) Dataset Characteristics with comparison (5) Availability and Sustainability (6) Conclusion and Future Work.

2 Relevance

Question Answering: Over the last years, KGQA systems are trying to evolve from a handcrafted rule based system to more robust machine learning(ML) based systems. Such ML approaches require large datasets for training and testing. For simple questions the KGQA community has reached a high level of accuracy but for more complex questions there is scope for much improvement. With a large scale dataset that incorporates a high degree of variety in the formal query expressions, provides a platform for machine learning models to improve the performance of KGQA with complex questions.

Solutions of NLP tasks using machine learning or semantic parsing have proved to be venerable to paraphrases. Moreover, if the system is exposed to paraphrases at the training period, the system could perform better and be more robust [1]. Thus having paraphrases of each original question enlarges the scope of the dataset.

[5] Qualifiers are used in order to further describe or refine the value of a property given in a fact statement: `https://www.wikidata.org/wiki/Help:Qualifiers`

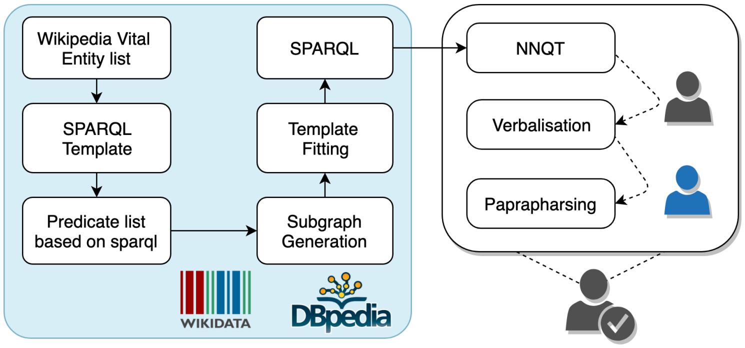

Fig. 1. Workflow for the dataset generation

Recently, DBpedia decided to adopt Wikidata's knowledge and mapping it to DBpedia's own ontology[7]. So far no dataset has based itself on this recent development. This work is the first attempt at allowing KGQA over the new DBpedia based on Wikidata [6].

Other Research Areas: *Entity and Predicate Linking*: This dataset may be used as a benchmark for systems which perform entity linking or/and relation linking on short text or on questions only. The previous version of the LCQuAD dataset has been used by such systems [5] and has enabled better performance of these modules.
SPARQL Query generation: The presented dataset has a high variety of SPARQL query templates which provides a use case for the modules which only focus on generating SPARQL given a candidate set of entities and relations.The SQG system [16] uses tree LSTMs to learn SPARQL generation and used the previous version of LCQuAD.
SPARQL to Natural language: This dataset may be used for natural language generation over knowledge graphs to generate complex questions at a much larger scale.

3 Dataset Generation Workflow

In this work the aim is to generate different varieties of questions at a large scale. Although different kinds of SPARQLs are used the corresponding natural language questions generated need to appear coherent to humans. Amazon Mechanical Turk (AMT) was used for generating the natural language questions from the system generated templates. A secondary goal is to make sure that the process of verbalisation of SPARQL queries on AMT does not require domain knowledge expertise of SPARQL and knowledge graphs on the part of the human workers (also known as turkers).

The core of the methodology is to generate SPARQL queries based on sparql templates, selected entities and suitable predicate. The SPARQLs are then transformed to Template Questions Q_T, which act as an intermediate stage between natural language

[6] at the time of writing this article, these updates do not reflect on the public DBpedia end-point. Authors have hosted a local endpoint of their own (using data from `http://downloads.dbpedia.org/repo/lts/wikidata/`). In future the authors shall release their own endpoint point with the new DBpedia model.

and formal language. Then a large crowd sourcing experiment(AMT) is conducted where the Q_Ts are verbalised to natural language questions - ie verbalised questions Q_V and then later paraphrase them to the paraphrased questions Q_P. To clarify, a Q_T instance represents SPARQL in a canonical structure which is human understandable. The generation of Q_T is a rule based operation.

The workflow is shown in the figure 1. The process starts with identifying a suitable set of entities for creating questions. A large set of entities based on Wikipedia Vital articles[7] is chosen and the corresponding same-as links to Wikidata IDs are found. Page-rank or entity popularity based approaches are avoided as it leads to dis-proportionately high number of entities from certain classes (say person). Instead Wikipedia Vital articles is chosen which provides important entities from a variety of topics such as people, geography, arts and several more, along with sub-topics. As a running example, say "Barack Obama" is selected from the list of entities.

Next a new set of SPARQL query templates are created such that they cover a large variety of question and intentions from a human perspective The template set is curated by observing other QA datasets and the KG architecture. All the templates have a corresponding SPARQL for Wikidata query end point and are valid on a DBpedia 2018 endpoint. The types of questions covered are as follows: simple question (1 fact), multiple fact question, questions that require additional information over a fact(wikidata qualifiers), temporal information question, two intention question and further discussed in Sec 4.3. Each class of questions also has multiple variations within the class.

Next, we select a predicate list based on the SPARQL template. For example if we want to make a "Count" question where user intends to know the number of times a particular predicate holds true, certain predicates such as "birthPlace" are disqualified as it will not make a coherent count-question. Thus different predicate white lists for different question types are maintained. Now the subgraph(fig 2) is generated from the KG based on the three factors - entity ("Barack Obama"), SPARQL template (say two intentions with qualifier), and a suitable predicate list. After slotting the predicate and sub-graph into the template the final SPARQL is generated. This SPARQL is then transformed to natural language templates, henceforth known as Q_T (Question Template), and the process is taken over by three step AMT experiments as discussed further.

The First AMT Experiment - Here the aim is to crowd-source the work of verbalising $Q_T \rightarrow Q_V$, where Q_V is the verbalisation of Q_T performed by a turker. Note that Q_T, since system generated, is often grammatically incorrect and semantically incoherrent, hence this step is required. For this we provided clear instruction to the turkers which vary according to the question type. For example: In two intention questions the turkers are instructed to make sure that none of the original intentions are missed in the verbalisation. Sufficient number of examples are provided to turkers so that they understand the task well. Again the examples vary according to the question type in the experiment.

The Second AMT Experiment - The task given to the turkers was to paraphrase the questions which have been generated in experiment 1, $Q_V \rightarrow Q_P$, where Q_P is a paraphrase of Q_V such that Q_P preserves the overall semantic meaning of Q_V while changing the syntactic content and structure. Turkers are encouraged to use synonyms, aliases and further changing the grammar structure of the verbalised question.

[7] https://en.wikipedia.org/wiki/Wikipedia:Vital_articles/Level/5

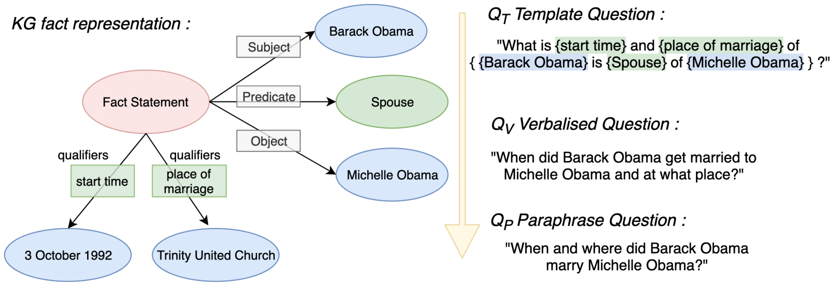
KG fact representation :
Subject
Barack Obama
Fact Statement
Predicate
Spouse
Object
Michelle Obama
qualifiers
qualifiers
start time
place of marriage
3 October 1992
Trinity United Church
Q_T Template Question :
"What is {start time} and {place of marriage} of { {Barack Obama} is {Spouse} of {Michelle Obama} } ?"
Q_V Verbalised Question :
"When did Barack Obama get married to Michelle Obama and at what place?"
Q_P Paraphrase Question :
"When and where did Barack Obama marry Michelle Obama?"

Fig. 2. (left) Representation of a fact with its Qualifiers. (right) Translation of a KG-fact to a verbalised question and then paraphrased question.

The Third AMT Experiment - This experiment performs human verification of experiments 1 and 2 and enforces quality control in the overall work flow. Turkers compare Q_T with Q_V and also Q_V to Q_P, to decide if the two pairs carry the same semantic meaning.The turkers are given a choice between "Yes / No / Can't say".

4 Dataset Characteristics

4.1 Dataset Statistics

In this section we analyse the statistics of our dataset. LCQuAD has 30,000 unique SPARQL - Question pairs. This dataset consists of 21,258 unique entities and 1,310 unique relations. Comparison of LCQuAD 2.0 to other related datasets is shown in the table 1. There are two datasets which cover simple questions, that is the question only requires one fact to answer. In this case the variation of formal queries is low. ComplexWebQuestion further extends the SPARQL of WebQuestions to generate complex questions. Though the number of questions in the dataset is in the same range as LCQuAD 2.0, the variation of SPARQLs is higher in LCQuAD 2.0 as it contains question 10 types question (such as boolean, dual intentions, Fact with qualifiers and other - ref 4.3) spread over 22 unique templates.

4.2 Analysis of Verbalisation and Paraphrasing experiments

To analyze the overall quality of verbalisation and paraphrasing by turkers we also used some automated methods (see figure 3). A good verbalisation of a system generated template ($Q_T \rightarrow Q_V$) would mean that Q_V preserves the semantic meaning of Q_T with the addition and removal of certain words. However a good paraphrasing of this verbalisation ($Q_V \rightarrow Q_P$) would mean that while the overall meaning is preserved, the order of words and also the words themselves (syntax) change to a certain degree. To quantify the sense of semantic-meaning vs change-of-word-order we calculate 1) cosine between vectors for each of these sentences pairs using BERT [4] embeddings - denoting "semantic similarity" 2) Levenshtein distance based syntax similarity between sentences showing the change in order of words.

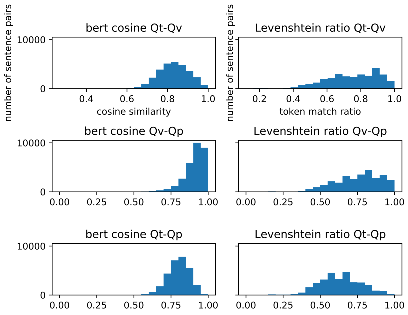

Fig. 3. Comparing Q_T , Q_V, Q_P based on the parameter (a.)Semantic Similarity and (b.) Syntactic Similarity

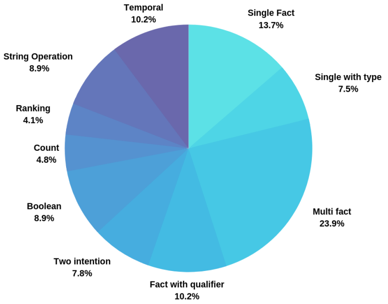

Fig. 4. Distribution of questions across all the question types

Data Set	Size	Variation	Query	Target KG	Paraphrase
Simple Questions[2]	100K	low	SPARQL	Freebase	No
30M Factoid Question[11]	30M	low	SPARQL	Freebase	No
QALD-9[10]	450	high	SPARQL	DBpedia	No
Free917[3]	917	medium	λ-Calculus	Freebase	No
WebQuestionSP[15]	5k	medium	SPARQL	Freebase	No
ComplexWebQuestionSP[12]	34K	medium	SPARQL	Freebase	No
LC-QuAD 1.0 [13]	5k	medium	SPARQL	DBpedia 2016-04	No
LC-QuAD 2.0	30K	high	SPARQL	Wikidata & DBpedia2018	Yes

Table 1. A comparison of datasets having questions and their corresponding logical forms

We observe that the cosine similarities of Q_T, Q_V and Q_P stay high (mean between 0.8 - 0.9 with standard deviation 0.07) denoting preservation of overall meaning throughout the steps, but syntax similarity stays comparatively low (mean between 0.6 - 0.75 with standard deviations between 0.14 to 0.16) since during verbalisation several words are added and removed from the imperfect system generated templates, and during paraphrasing the very task is to change the order of words of Q_V

The last set of histograms shows semantic similarity between Q_T and Q_P directly. Since we have skipped the verbalisation step in between we expect the distances to be farther away than other pairs. As expected the graphs show slightly lower cosine and syntax similarities than other pairs.

4.3 Types of Questions in LC-QuAD 2.0

1. Single fact: These queries are over a single fact(S-P-O). The query could return subject or object as answer. Example: "Who is the screenwriter of Mr. Bean?"
2. Single fact with type: This template brings type of constraint in single triple query. Example : "Billie Jean was on the tracklist of which studio album?"
3. Multi-fact: These queries are over two connected facts in Wikidata and have six

variations to them. Example:"What is the name of the sister city tied to Kansas City, which is located in the county of Seville Province?"
4. Fact with qualifiers: As shown in the fig. 2, qualifiers are additional property for a fact stored in KG. LC-QuAD 2.0 utilise qualifiers to make more informative questions. Such as "What is the venue of Barack Obama's marriage ?"
5. Two intention : This is a new category of query in KGQA, where the user question poses two intentions. This set of questions could also utilise the qualifier information as mentioned above and a two intention question could be generated, such as "Who is the wife of Barack Obama and where did he got married?" or "When and where did Barack Obama get married to Michelle Obama?".
6. Boolean : In boolean question, user intends to know if the given fact is true or false. LC-QuAD 2.0 not only generates questions which returns true by graph matching, but also generate false facts so that boolean question with "false" answers could be generated. We also use predicates that returns a number as an object, so that boolean questions regarding numbers could be generated. Example: "Did Breaking Bad have 5 seasons?"
7. Count : This set of questions uses the keyword "COUNT" in SPARQL, and performs count over the number of times a certain predicate is used with a entity or object. Example "What is the number of Siblings of Edward III of England ?"
8. Ranking : By using aggregates, we generate queries where the user intends an entity with maximum or minimum value of a certain property. We have three variations in this set of questions. Example :"what is the binary star which has the highest color index?"
9. String Operation: By applying string operations in SPARQL we generated questions where the user asks about an entity either at word level or character level. Example : "Give me all the Rock bands that starts with letter R ?"
10. Temporal aspect: This dataset covers temporal property in the question space and also in the answer space. A lot of the times facts with qualifiers poses temporal information. Example: "With whom did Barack Obama get married in 1992 ?"

5 Availability and Sustainability

To support sustainability we have published the dataset at figshare under CC BY 4.010 license. URL: `https://figshare.com/projects/LCQuAD_2_0/62270`

The repository of LC-QuAD 2.0 includes following files

- LC-QuAD 2.0 - A JSON dump of the Question Answering Dataset. (Test and Train)
- The dataset is available with Template question Q_T, Question Q_V, paraphrased question Q_P and corresponding SPARQLs for Wikidata and DBpedia. Other supplementary material to the dataset can be accessed from our website `http://lc-quad.sda.tech/`

6 Conclusion and Future Work

We presented the first large scale data set on Wikidata and upcoming DBpedia, consisting variety of complex questions. The dataset is generated in a semi-automatic setting that further requires crowd sourcing stages without domain knowledge expertise. In future we will maintain a benchmark strategy for KGQA systems on this dataset. We also plan to work towards developing a baseline KGQA system using the dataset LC-QuAD 2.0.

Acknowledgements This work has mainly been supported by the Fraunhofer-Cluster of Excellence “Cognitive Internet Technologies” (CCIT). It has also partly been supported by the German Federal Ministry of Education and Research (BMBF) in the context of the research project "InclusiveOCW" (grant no. 01PE17004D).

References

1. J. Berant and P. Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1415–1425, 2014.
2. A. Bordes, N. Usunier, S. Chopra, and J. Weston. Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075, 2015.
3. Q. Cai and A. Yates. Large-scale semantic parsing via schema matching and lexicon extension. In *ACL*, pages 423–433, 2013.
4. J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.
5. M. Dubey, D. Banerjee, D. Chaudhuri, and J. Lehmann. Earl: Joint entity and relation linking for question answering over knowledge graphs. In *International Semantic Web Conference*, pages 108–126. Springer, 2018.
6. M. Dubey, S. Dasgupta, A. Sharma, K. Höffner, and J. Lehmann. Asknow: A framework for natural language query formalization in sparql. In *International Semantic Web Conference*, pages 300–316, 2016.
7. A. Ismayilov, D. Kontokostas, S. Auer, J. Lehmann, S. Hellmann, et al. Wikidata through the eyes of dbpedia. *Semantic Web*, 9(4):493–503, 2018.
8. J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, and C. Bizer. DBpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *The Semantic Web*, pages 167–195, 2015.
9. D. Lukovnikov, A. Fischer, J. Lehmann, and S. Auer. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th International World Wide Web Conference*, pages 1211–1220, 2017.
10. N. Ngomo. 9th challenge on question answering over linked data (qald-9). *language*, 7:1.
11. I. V. Serban, A. García-Durán, Ç. Gülçehre, S. Ahn, S. Chandar, A. Courville, and Y. Bengio. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. In *54th Annual Meeting of the Association for Computational Linguistics*, 2016.
12. A. Talmor and J. Berant. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, 2018.
13. P. Trivedi, G. Maheshwari, M. Dubey, and J. Lehmann. Lc-quad: A corpus for complex question answering over knowledge graphs. In *International Semantic Web Conference*, pages 210–218. Springer, 2017.
14. D. Vrandečić and M. Krötzsch. Wikidata: a free collaborative knowledge base. 2014.
15. W.-t. Yih, M.-W. Chang, X. He, and J. Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on NLP*, 2015.
16. H. Zafar, G. Napolitano, and J. Lehmann. Formal query generation for question answering over knowledge bases. In *European Semantic Web Conference*. Springer, 2018.