VQuAnDa: Verbalization QUestion ANswering DAtaset

Endri Kacupaj[1 [0000−0001−5012−0420]], Hamid Zafar[1 [0000−0002−0407−1944]], Jens Lehmann[1,2 [0000−0001−9108−4278]], and Maria Maleshkova[1 [0000−0003−3458−4748]]

[1] University of Bonn, Bonn, Germany
{kacupaj,hzafarta,jens.lehmann,maleshkova}@cs.uni-bonn.de
[2] Fraunhofer IAIS, Dresden, Germany
jens.lehmann@iais.fraunhofer.de

Abstract. Question Answering (QA) systems over Knowledge Graphs (KGs) aim to provide a concise answer to a given natural language question. Despite the significant evolution of QA methods over the past years, there are still some core lines of work, which are lagging behind. This is especially true for methods and datasets that support the verbalization of answers in natural language. Specifically, to the best of our knowledge, none of the existing Question Answering datasets provide any verbalization data for the question-query pairs. Hence, we aim to fill this gap by providing the first QA dataset VQuAnDa that includes the verbalization of each answer. We base VQuAnDa on a commonly used large-scale QA dataset – LC-QuAD, in order to support compatibility and continuity of previous work. We complement the dataset with baseline scores for measuring future training and evaluation work, by using a set of standard sequence to sequence models and sharing the results of the experiments. This resource empowers researchers to train and evaluate a variety of models to generate answer verbalizations.

Keywords: Verbalization · Question Answering · Knowledge graph · Dataset.

Resource Type: Dataset
Website: http://vquanda.sda.tech/

Permanent URL: https://figshare.com/projects/VQuAnDa/72488

1 Introduction

Knowledge Graphs (KGs) have been gaining in popularity and adoption during the past years and have become an established solution for storing large-scale data, in both domain-specific (i.e., Knowlife [17]) and open-domain areas (i.e, Freebase [7], DBpedia [21] and Wikidata [34]). Despite the success of KGs, there are still some adoption hurdles that need to be overcome. In particular, users

need the expertise to use the formal query language supported by the KG in order to access the data within the KG. Question Answering (QA) systems aim to address this issue by providing a natural language-based interface to query the underlying KG. Thus, QA systems make KG data more accessible by empowering the users to retrieve the desired information via natural language questions rather than using a formal query language.

The early Knowledge Graph based Question Answering (KGQA) systems were mostly template or rule-based systems with limited learnable modules [32,14], mainly due to the fact that the existing QA datasets were small-scaled [9]. Consequently, researchers in the QA community are working on expanding QA datasets from two perspectives: (i) size: to support machine learning approaches that need more training data [8] and (ii) complexity: to move on from simple factoid questions to complex questions (e.g. multi-hop, ordinal, aggregation, etc) [6]. Note that while there are some QA datasets that are automatically generated [28], most QA datasets are manually created either by (i) using in-house workers [31] or crowd-sourcing [12] (ii) or extract questions from online question answering platforms such as search engines, online forum, etc [6]. The goal is to create datasets that are representative in terms of the types of questions that users are likely to ask.

These large-scale and complex QA datasets enable researchers to develop end-to-end learning approaches [25] and support questions with various features of varying complexity [1]. As a result, the main focus of many competitive QA methods is to enhance the performance of QA systems in terms of the accuracy of answer(s) retrieval. However, the average accuracy of the current state of the art QA approaches on manually created QA datasets is about 0.49, hence, there is plenty of room for improvement (See Figure 1).

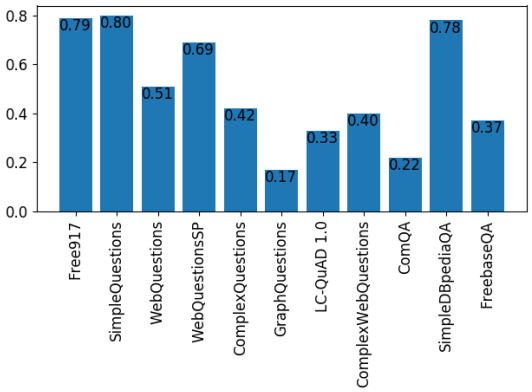

Fig. 1: The accuracy of the state of the art QA over KGs systems

Consequently, given this accuracy, the answers provided by a QA system need to be validated to assure that the questions are understood correctly and that the right data is retrieved. For instance, assuming that the answer to the exemplary question *"What is the longest river in Africa?"* is not known by the user. If the QA system only provides a name with no further explanation, the user might need to refer to an external data source to verify the answer. In an attempt to enable the users to verify the answer provided by a QA system, researchers employ various techniques such as (i) revealing the generated formal query [18], (ii) graphical visualizations of the formal query [36] and (iii) verbalizing the formal query [24,15,11]. We take a different approach to addressing the problem of validating the answers given by the QA system. We aim to verbalize the answer in a way that it conveys not only the information requested by the user but also includes additional characteristics that are indicative of how the answer was determined. For instance, the answer verbalization for the example question should be *"The longest river in Africa is Nile"* and given this verbalization, the user can better verify that the system is retrieving a *river* that is the *longest* river, which is located in *Africa*.

In this context we make the following contributions:

- We provide a framework for automatically generating the verbalization of answer(s), given the input question and the corresponding SPARQL query, which reduces the needed initial manual effort. The questions generated by the framework are subsequently manually verified to guarantee the accuracy of the verbalized answers.
- We present VQuAnDa – Verbalization QUestion ANswering DAtaset – the first QA dataset, which provides the verbalization of the answer in natural language.
- Evaluation baselines, based on a set of standard sequence to sequence models, which can serve to determine the accuracy of machine learning approaches used to verbalize the answers of QA systems.

The further advantages of having a dataset with accurate verbalization of the answer are multi-fold. Users do not need to understand RDF/ the formalization of the results, which decreases the adoption barriers of using KGs and QA systems. In addition, by providing indications of how the answer was derived as part of the verbalization, we enhance the explainability of the system. Furthermore, VQuAnDa serves as the basis for training and developing new ML models, which was up to date difficult due to the lack of data. Finally, our dataset lays the foundation for new lines of work towards extending VQuAnDa by the community.

The remainder of the paper is structured as follows. The next section presents the impact of our dataset within the QA community and its differences in comparison to the existing QA datasets. We introduce the details of our dataset and the generation workflow in Section 3. Section 4 discusses availability of the dataset, followed by the reusability study in Section 5, and Section 6 concludes our contributions.

2 Impact

Question Answering (QA) datasets over Knowledge Graphs (KG) commonly contain natural language questions, corresponding formal queries and/or the answer(s) from the underlying KG. Table 1 summarizes the features of all existing QA datasets over KGs. All QA datasets (except for [28]) are created using human annotators to ensure the quality of the results. In some datasets (such as FreebaseQA [20] and Free917 [9]), the questions are collected from search engines or other online question answering platforms and were subsequently adapted to an open-domain knowledge graph by human annotators. Others formulate the questions from a list of keywords (for instance LC-QuAD 1.0 [31]), or compose the question given a template-based automatically generated pseudo-question (for instance LC-QuAD 2.0 [12]).

Table 1: Summary of QA datasets over knowledge graphs

Dataset	KG	Size	Year	Formal Rep.	Creation
Free917 [9]	Freebase	917	2013	SPARQL	Manual
WebQuestions [6]	Freebase	5810	2013	None	Manual
SimpleQuestions [8]	Freebase	100K	2015	SPARQL	Manual
WebQuestionsSP [35]	Freebase	5810	2016	SPARQL	Manual
ComplexQuestions [5]	Freebase	2100	2016	None	Manual
GraphQuestions [29]	Freebase	5166	2016	SPARQL	Manual
30M Factoid Questions [28]	Freebase	30M	2016	SPARQL	Automatic
QALD (1-9) [3]	DBpedia	500	2011-2018	SPARQL	Manual
LC-QuAD 1.0 [31]	DBpedia	5000	2017	SPARQL	Manual
ComplexWebQuestions [30]	Freebase	33K	2018	SPARQL	Manual
ComQA [2]	Wikipedia	11K	2018	None	Manual
SimpleDBpediaQA [3]	DBpedia	43K	2018	Inferential Chain	Manual
CSQA [27]	Wikidata	200K	2018	Entities/Relations	Manual
LC-QuAD 2.0 [12]	Wikidata	30K	2019	SPARQL	Manual
FreebaseQA [20]	Freebase	28K	2019	Inferential Chain	Manual

The general trend in QA datasets is to work on the following aspects: (i) to increase the size of the dataset, (ii) to expand the question types to cover various features such as boolean queries, aggregations, ordinals in queries, etc. (iii) to increase the complexity of question by using compound features such as comparison or unions. Most recently, in an attempt to provide human-like conversations on a single topic, researchers expanded QA datasets to cover multiple utterances turns by introducing CSQA [27] – a sequential QA dataset in which instead of isolated questions, the benchmark contains a sequence of related questions along with their answers. However, the dataset contains only plain answers with no further verbalization to mimic human conversation.

On the one hand, recent advances in task-oriented dialogue systems resulted in releasing multi-turn dialogue datasets that are grounded through knowledge bases [16]. The intention is to provide training data for models allowing them to

have a coherent conversion. However, users cannot validate whether the provided answer at each step is correct. Moreover, the underlying knowledge graphs are significantly smaller than open-domain knowledge graphs such as DBpedia in terms of the number of entities and relations.

Considering the existing QA datasets and task-oriented dialogue datasets, we observe that the verbalization of answers with the intention to enable the users to validate the provided answer is neglected in the existing datasets. Consequently, the existing works cover either (i) the verbalization of the answer as in the dialog dataset, however, without empowering the users to validate the answer, (ii) or they enable the user to validate the answer, however, without a human like conversation [11].

We fill this gap in the question answering community by providing VQuAnDa, thus facilitating the research on semantic-enabled verbalization of answers in order to engage the users in human-like conversations while enabling them to verify the answers as well. We provide the verbalization of the answers by compiling all the necessary elements from the formal query and the answers into a coherent statement. Given this characterization of the answer, the user is enabled to verify whether the system has understood the intention of the question correctly. Furthermore, the dataset can be beneficial in dialog systems to not only hold the conversation but also to augment it with relevant elements that explain how the system comprehends the intention of the question.

We provide details on the dataset and the generation workflow in the next section.

3 VQuAnDa: Verbalization QUestion ANswering DAtaset

We introduce a new dataset with verbalized KBQA results called VQuAnDa. The dataset intends to completely hide any semantic technologies and provide a fluent experience between the users and the Knowledge Base. A key advantage of the verbalization is to support the answers given for a question/query. By receiving a complete natural language sentence as an answer, the user can understand how the QA system interpreted the question and what is the corresponding result. Table 2 shows some verbalization examples from our dataset. In the first example, the question *"What is the common school of Chris Marve and Neria Douglass?"* is translated to the corresponding SPARQL query, which retrieves the result `dbr:Vanderbilt_University` from the KB. In this case, the full verbalization of the result is *"[Vanderbilt University] is the alma mater of both Chris Marve and Neria Douglass."*. As it can be seen, this form of answer provides us the query result as well as details about the intention of the query.

Our dataset is based on the Largescale Complex Question Answering Dataset (LC-QuAD), which is a complex question answering dataset over DBpedia containing 5,000 pairs of questions and their SPARQL queries. The dataset was generated using 38 unique templates together with 5,042 entities and 615 predicates. To create our dataset, we extended the LC-QuAD by providing verbal-

Table 2: Examples from VQuAnDa

Question	What is the common school of Chris Marve and Neria Douglass?

Query	SELECT DISTINCT ?uri
	WHERE {
	dbr:Chris_Marve dbo:school ?uri .
	dbr:Neria_Douglass dbo:almaMater ?uri .
	}

Query result	dbr:Vanderbilt_University
Verbalization	[Vanderbilt University] is the alma mater of both Chris Marve and Neria Douglass.
Question	List all the faiths that British Columbian politicians follow?

Query	SELECT DISTINCT ?uri
	WHERE {
	?x dbp:residence dbr:British_Columbia .
	?x dbp:religion ?uri .
	?x a dbo:Politician .
	}

Query result	dbr:{Anglican, Anglicanism, Catholic Church, United Church of Canada, Fellowship of Evangelical Baptist Churches in Canada, Mennonite Brethren Church, story.html, Sikh, Roman Catholic}
Verbalization	The religions of the British Columbia politicians are [Anglican, Anglicanism, Catholic Church, United Church of Canada, Fellowship of Evangelical Baptist Churches in Canada, Mennonite Brethren Church, story.html, Sikh, Roman Catholic].

izations for all results. Furthermore, we improved the quality of the dataset by fixing grammar mistakes in the questions and, in some cases where the wording was unclear, completely rewriting them.

Given that Freebase is no longer publicly maintained, we decided to focus on the QA datasets that are based on other KGs such as DBpedia or Wikidata. Therefore, QALD, LC-QuAD 1.0, LC-QuAD 2.0 and CSQA are the only viable options. However, the size of the QALD dataset is significantly smaller in comparison to the other datasets (See Table 1). Moreover, in contrast to LC-QuAD 2.0 and CSQA that have not been yet used by any QA system, LC-QuAD 1.0 was the benchmarking dataset in more than 10 recent QA systems. Thus, we choose to built our dataset over LC-QuAD because of the large variety of questions and the manageable size that it has, which allows us to estimate the effectiveness of the produced results.

3.1 Generation Workflow

We followed a semi-automated approach to generate the dataset. The overall architecture of the approach is depicted in Figure 2.

Extract Results and Set Limit Initially, we retrieved the answers to all questions by using the DBpedia endpoint. Since some questions had multiple results,

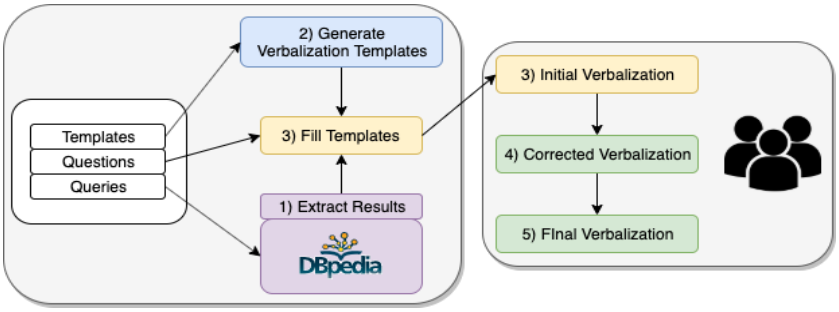

Fig. 2: Overview of dataset generation

we had to set a limit on how many answers will be shown as part of the verbalization. In the dataset, there are questions with one result and others with thousands. Creating a verbalization with a long list of all results is not intuitive, often not readable, and it is not contributing to the main focus of our work. Therefore we set a limit of a maximum of 15 results that are shown as part of the verbalization sentence. This limit was chosen by considering the different types of questions within the dataset and their complexity. In *Section 3.2 Statistics* we provide further details about the characteristics of the results. To handle the cases with more than 15 results we decided to replace them with an answer token ([*answer*]). For instance, for the question *"Which comic characters are painted by Bill Finger?"* the corresponding query retrieves 23 comic characters, therefore, the verbalization will include the answer token and it will be *"Bill Finger painted the following comic characters,* [*answer*]*."*. In this way, we can guarantee the sentence fluency for the particular example and we can still consider it for the verbalization task.

Generate Verbalization Templates Next, we generated the templates for the verbalized answers. In this step, we used the question templates from the LC-QuAD dataset. We decided to paraphrase them using a rule-based approach (see *Section Verbalization Rules*) and generate an initial draft version of the verbalizations.

Create Initial Verbalization In the following, we filled the templates with entities, predicates, and query results. To be able to distinguish the query results from the remaining parts of the verbalization sentence we decided to annotate them using box brackets. This provides us the flexibility whether we want to include, cover or exclude the results while working with the dataset.

Correct and Final Verbalizations While all initial draft versions of the verbalized answers were automatically generated, the last 2 steps had to be done manually in order to ensure the correctness of the verbalizations. First, we corrected and, if necessary, rephrased all answers to sound more natural and

fluent. Finally, to ensure the grammatical correctness of the dataset, we peer-reviewed all the generated results.

Verbalization Rules
During the generation workflow, we followed 4 rules on how to produce proper, fluent and correct verbalizations. These rules are:

- Use active voice;
- Use paraphrasing and synonyms;
- Construct the verbalization by using information from both the question and the query;
- Allow for rearranging the triple's order in verbalization.

The first and most important rule is the use of active voice as much as possible. In this way, we produce clean results that are close to human spoken language. The second rule is to paraphrase the sentences and use synonyms for generating different alternatives. The third rule is to base the verbalization on both questions and queries. We have many examples where the question is not directly related to a query from the aspect of structure and words it uses. During the process, we tried to balance out this difference by creating verbalizations that are closer to one or both of them. The last rule enables us to be flexible with the structure of the sentence. We tried not to directly verbalize the triple structure referred to by the query but also to shift the order of the subject and object in order to create more natural sounding sentences. All the rules have been heavily considered during the manual steps. For the automatic template generation, we mostly considered the first and last rule (active voice and sentence structure).

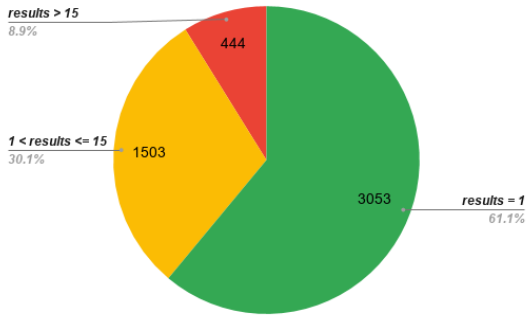

Fig. 3: Number of Results Returned per Query

3.2 Statistics

In this section we provide more details on the data contained in VQuAnDa, specifically focusing on the distribution of the query results. The dataset consists of 3053 (61.1%) questions that retrieve only one result from the knowledge

base. These examples include also boolean and count questions. There are 1503 (30.1%) examples that have more than one answer but less or equal to 15, which is the maximum number that we display as part verbalization. Finally, only 444 (8.9%) examples have more than 15 answers and are replaced with an answer token. Figure 3 depicts the result distribution.

Regarding the modified questions in the dataset – 340 (6.8%) examples in the LC-QuAD were revised to better represent the intention of the query. Some of the modifications are grammatical mistakes, while for others we had to completely restructure or even rewrite the questions. Figure 4 shows the number of modified questions, per question type and modification type.

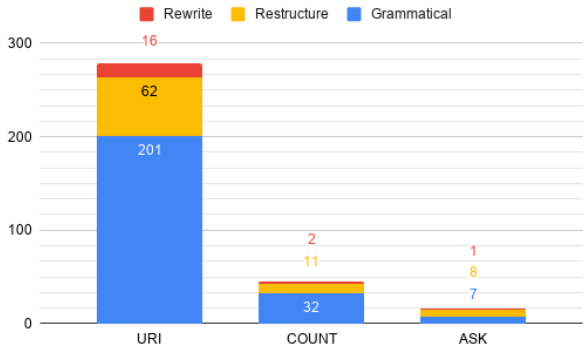

Fig. 4: Modified Questions in the Dataset

4 Availability and Sustainability

The dataset is available at AskNowQA[4] GitHub repository under the Attribution 4.0 International (CC BY 4.0) license. As a permanent URL, we also provide our dataset through figshare at `https://figshare.com/projects/VQuAnDa/72488`. The repository includes the training and test JSON files, where each of them contains the ids, questions, verbalized answers, and the queries.

The sustainability of the resource is guaranteed by the Question and Answering team of the Smart Data Analytics (SDA) research group at the University of Bonn and at Fraunhofer IAIS. A core team of 3 members is committed to taking care of the dataset, with a time horizon of at least 3 years. The dataset is crucial for currently ongoing PhD research and project work, and will, therefore, be maintained and kept up to date.

We are planning to have six-months release cycles, regularly updating the dataset based on improvement suggestions and corrections. However, we also plan to further extend VQuAnDa with more verbalization examples. We also aim to make the dataset a community effort, where researchers working in the

[4] `https://github.com/AskNowQA/VQUANDA`

domain of verbalization can update the data and also include their own evaluation baseline models. VQuAnDa should become an open community effort. Therefore, ESWC is the perfect venue for presenting and sharing this resource.

5 Reusability

The dataset can be used in multiple areas of research. The most suitable one is the knowledge base question answering area, since the initial purpose of the dataset was to support a more reliable QA experience. VQuAnDa allows researchers to train end-to-end models from generating the query, extracting the results and formulating the verbalization answer. Furthermore, the dataset can be used for essential QA subtasks such as entity and predicate recognition/linking, SPARQL query generation and SPARQL to question language generation. These subtasks are already supported by the LC-QuAD dataset. With the verbalizations, researchers can also experiment on tasks such as SPARQL to verbalized answer, question to verbalized answer or even hybrid approaches for generating better results. These possible lines of work indicate that the dataset is also useful for the natural language generation research area.

In summary, the use of the dataset is straightforward and allows researchers to further investigate different fields and discover other possible approaches were KBQA can be done more transparently and efficiently.

5.1 Experiments

To ensure the quality of the dataset but also to support its reuse we decided to perform experiments and provide some baseline models. These baseline models can be used as a reference point by anyone working with the dataset.

The experiments are done for the natural language generation task. We would like to test how easy it is for common neural machine translation or sequence to sequence models to generate the verbalized answers using as input only the question or the SPARQL query. To keep the task simple and because the answers appear only in the output verbalization part, we prefer to hide them with an answer token ($< ans >$). In this way, it will be enough for the model to predict only the position of the answer in the verbalization sentence.

We perform the experiments in two ways – i) the question or SPARQL query will be the input to our models and the expected output will be the correct verbalization; ii) we cover the entities in both input (question or query) and verbalization, so we allow the model to focus on other parts such as the sentence structure and the word relations. For the second experiment approach, we use the EARL framework [13] for recognizing the entities in both the question and answer sentences, and we cover them with an entity token ($< ent >$). For the queries, we can directly cover the entities, since we already know their positions. As we might expect, not all entities will be recognized correctly, but this can happen with any entity recognition framework and especially with datasets with complex sentences that contain one or multiple entities.

In the following subsections, we provide more details about the baseline models, the evaluation metrics, training details, and the results.

Baseline Models
For the baseline models, we decided to employ some standard sequence to sequence models. We first experiment with two RNN models that use different attention mechanisms [4,22]. For both RNN models we use bidirectional gated recurrent units (Bi-GRU) [10]. Next, we experiment with a convolutional sequence to sequence model, which is based on the original approach [19] where they employ a convolutional neural network (CNN) architecture for machine translation tasks. Finally, we use a transformer neural architecture, which is based on the original paper [33] where they create a simple attention-based sequence to sequence model.

Evaluation Metrics
BLEU: The first evaluation metric we use is the Bilingual Evaluation Understudy (BLEU) score introduced by [26]. The idea of the BLEU score is to count the n-gram overlaps in the reference; it takes the maximum count of each n-gram and it clips the count of the n-grams in the candidate translation to the maximum count in the reference. Essentially, BLEU is a modified version of precision to compare a candidate with a reference. However, candidates with a shorter length than the reference tend to give a higher score, while candidates that are longer are already penalized by the modified n-gram precision. To face this issue a brevity penalty (BP) was introduced, which is 1 if the candidate length c is larger or equal to the reference length r. Otherwise, the brevity penalty is set to $\exp(1 - r/c)$. Finally, a set of positive weights $\{w_1, ..., w_N\}$ is determined to compute the geometric mean of the modified n-gram precisions. The BLEU score is calculated by:

$$BLEU = BP \cdot \exp(\sum_{n=1}^{N} w_n \, log \, p_n), \tag{1}$$

where N is the number of different n-grams. In our experiments, we employ $N = 4$ and uniform weights $w_n = 1/N$.

Perplexity: To estimate how well our models predict the verbalization we are using the perplexity metric. For measuring the similarity of a target probability distribution p and an estimated probability distribution q, we are using cross entropy $H(p, q)$ which is defined by

$$H(p, q) = -\sum_{x} p(x) \, log \, q(x), \tag{2}$$

where x indicates the possible values in the distribution. The perplexity is defined as the exponentiation of cross entropy:

$$Perplexity(p, q) = 2^{H(p,q)}. \tag{3}$$

In our case, the target distribution p is the encoding vector of the verbalization vocabulary and q is the prediction output of the decoder. We calculate perplexity after every epoch using the averaged cross entropy loss of the batches.

Researchers have shown [23] that perplexity strongly correlates with the performance of machine translation models.

Training
To keep the comparison fair across the models we employ the same training parameters for all. We split the data into 80-10-10 where 80% is used for training, 10% for validation and the last 10% for testing. The batch size is set to 100 and we train for 50 epochs. During the training, we save the model state with the lowest loss on the validation data.

We tried to keep the models almost of the same size regarding their trainable parameters. More precisely, for the first RNN model we use an embedding dimension of 256, the hidden dimension is 512 and we use 2 layers. We also apply dropout with probability 0.5 on both encoder and decoder. For the second RNN model, we keep everything the same except the embedding dimension where we decided to double it to 512. For the convolutional model, we set the embedding dimension to 512, we keep all the channels in the same dimension of 512 and we use a kernel size of 3. We use 3 layers for the encoder and decoder. Similar to RNNs, the dropout here is set to 0.5. Finally, for the transformer model, the embedding dimension is 512, we use 8 heads and 2 layers. The dropout here is set to 0.1. For the first two RNN models we use a teacher forcing value of 1.0 so we can compare the results with the other models.

We do not use any pretrained embedding model. For building the vocabularies we use a simple one-hot encoding approach. For all the models we use Adam optimizer, and cross entropy as a loss function. All our experiments are publicly available here `https://github.com/endrikacupaj/VQUANDA-Baseline-Models`.

Table 3: Perplexity experiment results

		With Entities		Covered Entities	
Models	Input	Validation	Test	Validation	Test
RNN-1 [4]	Question	8.257	8.865	5.709	5.809
	Query	6.823	7.029	5.212	5.335
RNN-2 [22]	Question	8.494	8.802	5.799	5.891
	Query	6.727	6.999	5.259	5.394
Convolutional [19]	Question	**4.137**	**4.194**	**3.409**	**3.451**
	Query	**3.175**	**3.311**	**3.158**	**3.201**
Transformer [33]	Question	5.232	5.464	3.716	3.727
	Query	3.978	4.062	3.229	3.292

Results
Beginning with the perplexity results, in Table 3 we can see that the convolutional model outperforms all other models and is considered the best. The transformer model comes second and is pretty close to the convolutional. The RNN models perform considerably worse comparing the other two.

Since perplexity is the exponentiation of cross entropy, the lower the value the better the results, which means that the best possible value is 1. The convo-

lutional model using the question as input achieves 4.1 with entities and 3.4 with covered entities on validation and test data. When we use the SPARQL query as input the perplexity gets a lower value, which means the model performs slightly better. In particular, for the convolutional model, we obtain 3.3 with entities and 3.2 with covered entities on test data. The improved performance using the query as input is expected since the model receives the same pattern of queries every time.

Table 4: BLEU score experiment results

		With Entities		Covered Entities	
Models	Input	Validation	Test	Validation	Test
RNN-1 [4]	Question	14.00	12.86	25.09	24.88
	Query	18.40	17.76	30.74	29.25
RNN-2 [22]	Question	15.53	15.43	27.63	26.95
	Query	22.29	21.33	**34.34**	30.78
Convolutional [19]	Question	**21.49**	**21.30**	**28.21**	**27.73**
	Query	**26.02**	**25.95**	32.61	**32.39**
Transformer [33]	Question	19.00	18.38	25.67	26.58
	Query	24.16	22.98	31.65	29.14

In any case, there is still a lot of space for improvement until we can say that the task is solved. The BLEU score further supports this fact. By looking at Table 4 with the BLEU score results we can see that again the convolutional model performs best with a value of up to 32 with covered entities. Without covering entities the best we get on test data is almost 26, which is not really an adequate result. The best possible value for the BLEU metric is 100. A score of more than 60 is considered a perfect translation that often outperforms humans. For our dataset, there is still a lot of research required until we produce models that can reach these numbers.

5.2 Use by the Community

Currently, we are actively developing and sharing the dataset within the scope of two projects – SOLIDE and CLEOPATRA. The work is very well received, however, our ultimate goal is to make VQuAnDa an open community effort.

SOLIDE [5] In major disastrous situations such as flooding, emergency services are confronted with a variety of information from different sources. The goal of the SOLIDE project is to analyze and process the information from multiple sources in order to maintain a knowledge graph that captures an overall picture of the situation. Furthermore, using a voice-based interface, users can ask various questions about the ongoing mission, for instance, *"How many units are still available?"*. Given the dangerous circumstances, it is vital to assert the user that the provided answer is complete and sound. Hence, the system needs to

[5] `http://solide-projekt.de`

verbalize its internal representation of the question (e.g. SPARQL) with the answer as *"There are 2 units with the status available"*. Thus, VQuAnDa is essential for being able to learn a verbalization model as part of the solution framework of the SOLIDE project.

CLEOPATRA ITN [6] As European countries become more and more integrated, an increasing number of events, such as the Paris shootings and Brexit, strongly affect the European community and the European digital economy across language and country borders. As a result, there is a lot of event-centric multilingual information available from different communities in the news, on the Web and in social media. The Cleopatra ITN project aims to enable effective and efficient analytics of event-centric multilingual information spread across heterogeneous sources and deliver results meaningful to the users. In particular in the context of question answering, Cleopatra advances the current state of the art by enabling user interaction with event-centric multilingual information. Considering this challenge, the VQuAnDa dataset serves as a basis for learning a question answering model, while the verbalizations are employed to enhance the interactivity of the system.

In addition to using the dataset in order to conduct research and enable the work within projects, we also use it for teaching purposes. VQuAnDa and pre-trained models are given to the students so that they can try out machine learning approaches by themselves and evaluate the produced results by looking at the quality of the generated verbalizations. While the dataset already has a solid level of reuse, we see great potential for further adoption by the Semantic Web community, especially in the areas of applied and fundamental QA research.

6 Conclusion and Future Work

We introduce VQuAnDa – the first QA dataset including the verbalizations of answers in natural language. We complement the dataset by a framework for automatically generating the verbalizations, given the input question and the corresponding SPARQL query. Finally, we also share a set of evaluation baselines, based on a set of standard sequence to sequence models, which can serve to determine the accuracy of machine learning approaches used to verbalize the answers of QA systems. Without a doubt, the dataset presents a very valuable contribution to the community, providing the foundation for multiple lines of research in the QA domain.

As part of future work, we plan to develop an end-to-end model that will use the question to obtain the correct answer and at the same time to generate the correct verbalization. Moreover, we would like to focus on the verbalization part by researching possible models that can improve the results. The baseline models we used for the dataset receive as input the question or the query. We assume that a hybrid approach can make the model benefit from both input types and possibly produce better results. Finally, we will also work on continuously extending and improving VQuAnDa.

[6] `http://cleopatra-project.eu/`

Acknowledgments

This work was supported by the European Union H2020 founded project CLEOPATRA (ITN, GA. 812997).

References

1. Abdelkawi, A., Zafar, H., Maleshkova, M., Lehmann, J.: Complex query augmentation for question answering over knowledge graphs. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (2019)
2. Abujabal, A., Roy, R.S., Yahya, M., Weikum, G.: Comqa: A community-sourced dataset for complex factoid question answering with paraphrase clusters. arXiv preprint arXiv:1809.09528 (2018)
3. Azmy, M., Shi, P., Lin, J., Ilyas, I.: Farewell freebase: Migrating the simplequestions dataset to dbpedia. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 2093–2103 (2018)
4. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. arXiv e-prints arXiv:1409.0473 (Sep 2014)
5. Bao, J., Duan, N., Yan, Z., Zhou, M., Zhao, T.: Constraint-based question answering with knowledge graph. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 2503–2514 (2016)
6. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1533–1544 (2013)
7. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data (2008)
8. Bordes, A., Usunier, N., Chopra, S., Weston, J.: Large-scale simple question answering with memory networks. arXiv preprint arXiv:1506.02075 (2015)
9. Cai, Q., Yates, A.: Large-scale semantic parsing via schema matching and lexicon extension. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (2013)
10. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv e-prints arXiv:1406.1078 (Jun 2014)
11. Diefenbach, D., Dridi, Y., Singh, K., Maret, P.: Sparqltouser: Did the question answering system understand me? (2017)
12. Dubey, M., Banerjee, D., Abdelkawi, A., Lehmann, J.: Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In: International Semantic Web Conference. pp. 69–78. Springer (2019)
13. Dubey, M., Banerjee, D., Chaudhuri, D., Lehmann, J.: Earl: Joint entity and relation linking for question answering over knowledge graphs. In: ISWC 2018
14. Dubey, M., Dasgupta, S., Sharma, A., Höffner, K., Lehmann, J.: Asknow: A framework for natural language query formalization in sparql. In: ESWC (2016)
15. Ell, B., Harth, A., Simperl, E.: Sparql query verbalization for explaining semantic search engine queries. In: European Semantic Web Conference (2014)
16. Eric, M., Manning, C.D.: Key-value retrieval networks for task-oriented dialogue. arXiv preprint arXiv:1705.05414 (2017)
17. Ernst, P., Meng, C., Siu, A., Weikum, G.: Knowlife: a knowledge graph for health and life sciences. In: 2014 IEEE 30th International Conference on Data Engineering

18. Ferré, S.: Sparklis: an expressive query builder for sparql endpoints with guidance in natural language. Semantic Web **8**(3), 405–418 (2017)
19. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional Sequence to Sequence Learning. arXiv e-prints arXiv:1705.03122 (May 2017)
20. Jiang, K., Wu, D., Jiang, H.: Freebaseqa: A new factoid qa data set matching trivia-style question-answer pairs with freebase. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics
21. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web Journal (2015)
22. Luong, M.T., Pham, H., Manning, C.D.: Effective Approaches to Attention-based Neural Machine Translation. arXiv e-prints arXiv:1508.04025 (Aug 2015)
23. Luong, T., Kayser, M., Manning, C.D.: Deep neural language models for machine translation. In: Proceedings of the 19th Conference on Computational Natural Language Learning. Association for Computational Linguistics, Beijing, China (2015)
24. Ngonga Ngomo, A.C., Bühmann, L., Unger, C., Lehmann, J., Gerber, D.: Sparql2nl: verbalizing sparql queries. In: Proceedings of the 22nd International Conference on World Wide Web. pp. 329–332. ACM (2013)
25. Nilesh Chakraborty, Denis Lukovnikov, G.M.P.T.J.L.A.F.: Introduction to neural network based approaches for question answering over knowledge graphs (2019)
26. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (2002)
27. Saha, A., Pahuja, V., Khapra, M.M., Sankaranarayanan, K., Chandar, S.: Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In: Thirty-Second AAAI Conference (2018)
28. Serban, I.V., García-Durán, A., Gulcehre, C., Ahn, S., Chandar, S., Courville, A., Bengio, Y.: Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. arXiv preprint arXiv:1603.06807 (2016)
29. Su, Y., Sun, H., Sadler, B., Srivatsa, M., Gur, I., Yan, Z., Yan, X.: On generating characteristic-rich question sets for qa evaluation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (2016)
30. Talmor, A., Berant, J.: The web as a knowledge-base for answering complex questions. arXiv preprint arXiv:1803.06643 (2018)
31. Trivedi, P., Maheshwari, G., Dubey, M., Lehmann, J.: Lc-quad: A corpus for complex question answering over knowledge graphs. In: International Semantic Web Conference. pp. 210–218. Springer (2017)
32. Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.C., Gerber, D., Cimiano, P.: Template-based question answering over rdf data. In: Proceedings of the 21st international conference on World Wide Web. pp. 639–648. ACM (2012)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need. arXiv e-prints arXiv:1706.03762
34. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledge base (2014)
35. Yih, W.t., Richardson, M., Meek, C., Chang, M.W., Suh, J.: The value of semantic parse labeling for knowledge base question answering. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (2016)
36. Zheng, W., Cheng, H., Zou, L., Yu, J.X., Zhao, K.: Natural language question/answering: Let users talk with the knowledge graph. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (2017)