Would You Ask it that Way? Measuring and Improving Question Naturalness for Knowledge Graph Question Answering

Trond Linjordet
University of Stavanger
Stavanger, Norway
trond.linjordet@uis.no

Krisztian Balog
University of Stavanger
Stavanger, Norway
krisztian.balog@uis.no

ABSTRACT

Knowledge graph question answering (KGQA) facilitates information access by leveraging structured data without requiring formal query language expertise from the user. Instead, users can express their information needs by simply asking their questions in natural language (NL). Datasets used to train KGQA models that would provide such a service are expensive to construct, both in terms of expert and crowdsourced labor. Typically, crowdsourced labor is used to improve template-based pseudo-natural questions generated from formal queries. However, the resulting datasets often fall short of representing genuinely natural and fluent language. In the present work, we investigate ways to characterize and remedy these shortcomings. We create the IQN-KGQA test collection by sampling questions from existing KGQA datasets and evaluating them with regards to five different aspects of naturalness. Then, the questions are rewritten to improve their fluency. Finally, the performance of existing KGQA models is compared on the original and rewritten versions of the NL questions. We find that some KGQA systems fare worse when presented with more realistic formulations of NL questions. The IQN-KGQA test collection is a resource to help evaluate KGQA systems in a more realistic setting. The construction of this test collection also sheds light on the challenges of constructing large-scale KGQA datasets with genuinely NL questions.

CCS CONCEPTS

• **Information systems** → **Question answering**; **Test collections**.

KEYWORDS

Knowledge graph question answering; test collections; question naturalness

ACM Reference Format:
Trond Linjordet and Krisztian Balog. 2022. Would You Ask it that Way? Measuring and Improving Question Naturalness for Knowledge Graph Question Answering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3477495.3531739

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531739

Table 1: Example questions, each rewritten by crowd workers as a more natural way to express the original question.

Original question (DBNQA [12])	**Rewritten question** (IQN-KGQA [this paper])
List the territory of romanian war of independence ?	What territory was involved in the Romanian War of Independence?
Name the nearest city to la laguna lake ?	What is the nearest city to La Laguna Lake?
What is the government type of wallis and futuna ?	What type of government does Wallis and Futuna have?
What is the origin of faber-rebe?	What is the origin of the faber-rebe grape?
What is the total number of writers whose singles are recorded in ferndale?	How many writers had singles recorded in Ferndale?

1 INTRODUCTION

Knowledge Graph Question Answering (KGQA) is an approach to answering users' questions that both harnesses structured data in the form of knowledge graphs (KGs) and also allows the user to articulate their information need in natural language (NL). Training machine learning models for KGQA requires large-scale datasets specific to the KGQA task. Most commonly, such datasets consist of instances that each comprises a formal query (also known as logic form) and a corresponding NL question [15].

In order to construct large KGQA datasets, the work is typically divided into expert and non-expert subtasks which are then assigned to different people. This makes sense economically, but the resulting dataset may have qualitative shortcomings as a result. The formal query is typically constructed by experts or generated synthetically, while the NL questions are typically added by crowdsourced labor tasked with paraphrasing some generated pseudo-natural form of the corresponding formal query [30]. The NL question is thus typically not formulated by the same person who devised the formal query. Critically, this decouples the intent of the formal query from the NL question meant to express that intent. In addition, in large-scale dataset construction, the data is often back-generated from formal queries, that is, the formal query is generated based on available data, and the corresponding NL question is created afterwards. An individual working with a KG for practical reasons would first develop an information need, which may or may not be first expressed as an NL question, and only then construct a formal query to represent that information need. The crowd worker is also not guaranteed to be completely fluent in the specific language that is used in the dataset being constructed. Furthermore, even so-called open-domain KGQA typically consists of questions in a variety of specific domains. If the crowd worker is unfamiliar with this domain, they may not be able to apply the appropriate wording for the underlying domain and categories. We therefore hypothesize that this approach to KGQA dataset construction does not ensure genuinely natural NL questions.

In Table 1 we have listed three example questions sampled from existing KGQA datasets, each in their original form and in a rewritten form, generated by additional rounds of crowdsourced paraphrasing and quality control. These are all examples where a KGQA model trained on the original dataset performed perfectly on the original question, but completely failed on the rewritten question. This illustrates that some KGQA systems trained on less natural NL questions are not able to address a more naturally phrased version of the same question.

From a machine learning perspective, it is unsurprising that test data from a different distribution than training data may be challenging. However, as the rewritten questions in Table 1 illustrate, the KGQA models are failing on more naturally articulated questions. This calls into question whether KGQA models are really learning to perform their nominal task. We investigate how NL questions in KGQA datasets can be considered unnatural, and develop a coding scheme for dimensions of unnaturalness. We determine five dimensions of unnaturalness in NL questions: grammar, form, meaning, answerability, and factuality.

Next, we use our coding scheme in a crowdsourcing context to characterize original NL questions and collect rewritten forms of these NL questions, which are included in our test collection, IQN-KGQA. We sample 250 NL questions from each of three benchmark KGQA datasets: DBNQA [12], LC-QuAD v2.0 [9], and GrailQA [11].

To develop truly effective KGQA systems requires an appreciation of how well these systems fare against realistic questions formulated in genuinely natural language. We apply KGQA models to the original and rewritten questions and see how improved naturalness challenges existing systems. We find that performance drops up to 78% when KGQA models are challenged with the set of rewritten questions.

The novel contributions of this work include:

- A novel coding scheme to characterize to what extent nominal natural language questions in KGQA datasets actually constitute natural language.
- Experimental designs for measuring question naturalness (to flag questions that are unnatural) and improving question fluency and composition using crowdsourcing.
- A novel test collection, IQN-KGQA, consisting of 3x250 questions sampled from 3 prominent KGQA datasets, made publicly available at https://github.com/iai-group/IQN-KGQA. The sampled questions are rated on naturalness along 5 dimensions by at least 3 crowd workers each, and rewritten for greater naturalness where possible.
- A comparison of existing KGQA models on original and rewritten questions.

2 RELATED WORK

The field of KGQA is in many ways defined by the datasets used to train and test systems. In the following, we describe some salient milestone KGQA datasets and their manner of construction. Previous work [7, 15, 19, 31] has surveyed the field of KGQA, which we draw on in our present summary. The KGQA datasets are grounded in one or more of the three most common open-domain knowledge graphs (KGs): Freebase, DBpedia, and Wikidata. We note that the overall trend in KGQA dataset construction has been towards more complex formal queries as well as larger datasets. For each dataset, we defer to the respective papers' stance as to whether the dataset should be considered to contain complex formal queries.

Cai and Yates [6] create the dataset Free917 by asking two native English speakers to ask questions in multiple domains, and then annotating these questions with formal queries.

Berant et al. [4] construct the dataset WebQuestions, consisting of 5810 instances with only NL questions and answers, but no formal queries. The dataset is constructed by generating single-entity questions with the Google Suggest API, and then crowdsourcing answers based only on the Freebase page of the entity in a given NL question. Question-answer pairs are kept as instances when at least two crowd workers agree on an answer.

Bordes et al. [5] create the large dataset SimpleQuestions, consisting only of NL questions that can be answered by a single fact (SPO-triple) in the KG, and the corresponding fact. The dataset is constructed by shortlisting a set of facts, and then having English-speaking annotators generate NL questions mentioning the subject and predicate of the fact, such that the answer would be the object.

Bao et al. [3] construct the dataset ComplexQuestions consisting of question-answer pairs by mining a search query log for search queries with overlapping terms as in WebQuestions and SimpleQuestions, and then categorize the search queries according to some rules to identify multi-constraint questions. The questions are manually annotated with answers. Additional question-answer pairs are taken directly from pre-existing datasets.

Su et al. [24] construct the dataset GraphQuestions—where each instance includes NL question, formal query, and ground truth answer—by first generating query graphs, and then converting these to NL questions via crowdsourcing. The ground truth answer is retrieved by converting the query graph to a formal query and executing it. This approach to crowdsourcing for KGQA datasets has been referred to as the Overnight method [24].

Yih et al. [32] construct the dataset WebQuestionsSP by having experts annotate instances in WebQuestions [4] with SPARQL queries where feasible.

Talmor and Berant [25] construct the dataset ComplexWebQuestions by programmatically generating more complex formal queries from WebQuestionsSP, then generating pseudo-NL questions for crowd workers to improve into NL questions.

The QALD series (1–9) [17] consists of small datasets of questions generated by students and formal queries hand-crafted by experts.

Trivedi et al. [26] construct the dataset LC-QuAD v1.0, which consists of NL questions and formal queries. First query graph templates are combined with whitelisted (non-metadata) entities and predicates to instantiate specific formal queries, then pseudo-NL

Table 2: Overview of KGQA datasets. Those marked with † are considered in our study.

Dataset	KG	Size	Crowdsourcing
Free917 [6]	Freebase	917	Unclear
WebQuestions [4]	Freebase	5,810	Yes
SimpleQuestions [5]	Freebase	108,442	Unclear
ComplexQuestions [3]	Freebase	2,100	No
GraphQuestions [24]	Freebase	5,166	Yes
WebQuestionsSP [32]	Freebase	4,737	No
ComplexWebQuestions [25]	Freebase	34,689	Yes
QALD series (1–9) [17, 27]	DBpedia	∼50-500 each	No
LC-QuAD v1.0 [26]	DBpedia	5,000	No
DBNQA [12]†	DBpedia	894,499	No
LC-QuAD v2.0 [9]†	DBpedia, Wikidata	30,000	Yes
CFQ [14]	Freebase	239,357	No
GrailQA [11]†	Freebase	64,331	Yes
KQA Pro [21]	Wikidata	117,970	Yes

questions are generated from the formal queries, which are corrected or paraphrased in two rounds with independent annotators.

Hartmann et al. [12] construct the dataset DBNQA, consisting of NL questions and formal queries, from the LC-QuAD v1.0 [26] and QALD-7-train [27] datasets. The extant datasets are taken as the basis to extract templates for both formal queries and NL questions, and those templates are then instantiated with different entity and predicate bindings. DBNQA* [16] partitions DBNQA [12] into training, validation, and test splits based on the underlying templates, avoiding leakage of information between training and test splits. The instances are identical to DBNQA, and so we use DBNQA* in our experiments.

Dubey et al. [9] construct the dataset LC-QuAD v2.0, extending the workflow established by Trivedi et al. [26] by crowdsourcing the paraphrasing of generated pseudo-NL questions into improved NL questions. This also includes several rounds of crowd workers generating further paraphrasing of NL questions and performing quality control on others' annotations.

Keysers et al. [14] construct the dataset CFQ, with instances comprising formal queries and and NL questions, in a completely rules-based manner.

Gu et al. [11] construct the dataset GrailQA following the Overnight [24] approach of generating formal queries and pseudo-NL questions, and then using crowdsourcing to paraphrase pseudo-NL questions into NL questions, and finally using crowd workers to cross-validate the paraphrases of their colleagues.

Shi et al. [21] construct the dataset KQA Pro in a similar manner as Gu et al. [11], including the use of crowdsourced labor for paraphrasing pseudo-NL questions and cross-validation.

From these examples of KGQA dataset construction, summarized in Table 2, we see that crowdsourcing is commonly used with the intent of paraphrasing pseudo-NL questions into more genuine NL questions.

3 PRELIMINARY ANALYSIS

Larger KGQA datasets typically rely on crowdsourcing for generating NL questions from synthetically generated formal queries. We hypothesize that scaling up a KGQA dataset by relying heavily on these two distinct modes comes at the expense of NL question quality, and that the original NL questions may not always be genuinely natural NL questions. To investigate unnaturalness in the NL questions of existing KGQA datasets, we begin by testing that hypothesis on a small sample of instances using expert annotators.

We select three KGQA datasets to sample NL questions from. We choose KGQA datasets that are recent, large, have complex questions and formal queries. We also choose the datasets so that all of the most common KGs are represented in the formal query bindings. Specifically, we consider the datasets DBNQA* [16], LC-QuAD v2.0 [9], and GrailQA [11]. We then randomly sample 25 NL questions from each of these datasets. Specifically, the 25 NL questions are respectively sampled from the entire DBNQA* dataset, and from the train splits of LC-QuAD v2.0 and GrailQA.

Following the approaches of Arguello et al. [1] and Jørgensen and Bogers [13], we perform an *open coding* pass to collect impressions on how the NL questions fall short of being "natural." Three academic researchers are presented each NL question and asked to (i) judge whether or not the question is natural, (ii) produce a (more) natural paraphrase of the question, and (iii) comment on the NL question and suggest any tags or categories regarding "why and how the question is or is not natural." The first author then collates the responses, and the comments and categories are harmonized into a consistent coding scheme of tags by the first author. Both authors review the extracted tags and discuss common themes across tags. The tags are then organized into the five dimensions of unnaturalness illustrated along with NL question examples in Table 3. We note that the examples in the Table may exhibit more than one of the properties the exemplify a given tag or dimension of unnaturalness.

Table 3: Dimensions of question unnaturalness

Dimension	Tag	Example
Grammar	Grammatical errors	Which is {godmother} of {Camillo Benso di Cavour}, whose {craft} is {politician} ?
	Poor flow/word ordering	Who lives in Anita Bryant whose arrondissement is Pittsburg County?
	Non-idiomatic	What is character role of Turandot ?
Form	Quizlike	astronaut gerhard thiele is associated with which space agency?
	Imperative	find beaufort wind force whose wave height is 0.1
	Inconcise	Which is the regression analysis that is used by the logistic regression analysis and contains the word logistic in it's name?
Meaning	Inconsistent domains/categories	Was 6063 jason invented in eugene merle shoemaker
	Overly specific	Which university attended by arturo macapagal was also the alma mater of hector tarrazona ?
	Redundant constraint	What is the death place of the étienne pélabon and is the birthplace of the abeille de perrin?
Answerability	Under-constrained	which organism was born on 1926-06?
	Nonsense/Unintelligible	what routed drug that a marketed formulation that has a reference form of neurontin 250 solution?
Factuality	Two questions	Who was married to Faye Dunaway and when did it end?
	Descriptive answer expected	What is a crescent?

4 DATA ANNOTATION

Having defined codes to characterize question unnaturalness in KGQA datasets, we next design a protocol for larger-scale data labeling using a two-step crowdsourcing pipeline. Crowd workers are first asked to annotate and paraphrase the sampled NL questions. Then, in a separate task, a different set of workers is employed to select the best version of a question from a set, including the original formulations as well as rewritten questions from the first task.

We sample a new set of NL questions, this time 250 NL questions from each of the three datasets. Specifically, we randomly sample the NL questions from the test split of DBNQA* and LC-QuAD v2.0, but from the validation split of GrailQA, since for the latter, the public test split does not include ground truth answers. The resulting test collection is termed IQN-KGQA and is summarized in Table 4.

Table 4: Summary of the IQN-KGQA collection.

Subset	Split	#Questions	#Rewritten
DBNQA*	test	250	180
LC-QuAD v2.0	test	250	150
GrailQA	validation	250	211
Total		750	541

4.1 Crowdsourcing: Platform and Workers

Our data annotation was conducted on the Amazon Mechanical Turk platform. For both tasks, workers were required to have a HIT approval rate of 98% with more than 1000 approvals. The payments were set to USD $0.30 and $0.15, respectively, based on the estimated effort demanded for each task.

Crowd workers were not required to have domain knowledge, based on the findings of Dubey et al. [9]. Since only open-domain KGQA datasets are used in the present work, the annotation tasks are designed to rely on common sense and English language knowledge primarily. For example, the prompt for the Likert scale "answerablility" is the question "Would you be able to answer this question with the help of a search engine or Wikipedia?" In other words, the data annotation relies on metacognition with respect to an NL question rather than actually finding some answer. Identifying crowd workers with comparable levels of expertise in specific domains prior to data annotation would present a major additional cost. Also, those workers would not necessarily be representative of the general user population whose information needs KGQA datasets aim to capture.

4.2 Task 1: Annotate and Rewrite

In the first task, the crowd workers are given one of the sampled NL questions (the target question) and are asked to rate the question in terms of the five dimensions of unnaturalness. For each dimension, the question is rated on a Likert scale. Next, the crowd workers are asked to rewrite the question, to "write a better, more natural and correct version" of the question. Finally, the crowd workers are asked to indicate if they rewrote the question, and if not what the reason was, including a free text field to elaborate on any "other" reason for not rewriting. The complete form and instructions are provided in the GitHub repository accompanying this paper.

The responses from crowd workers are then quality controlled, and responses which overtly demonstrate a lack of genuine effort are entirely removed. Criteria for this exclusion include indicating that a question was rewritten but providing no paraphrase, writing

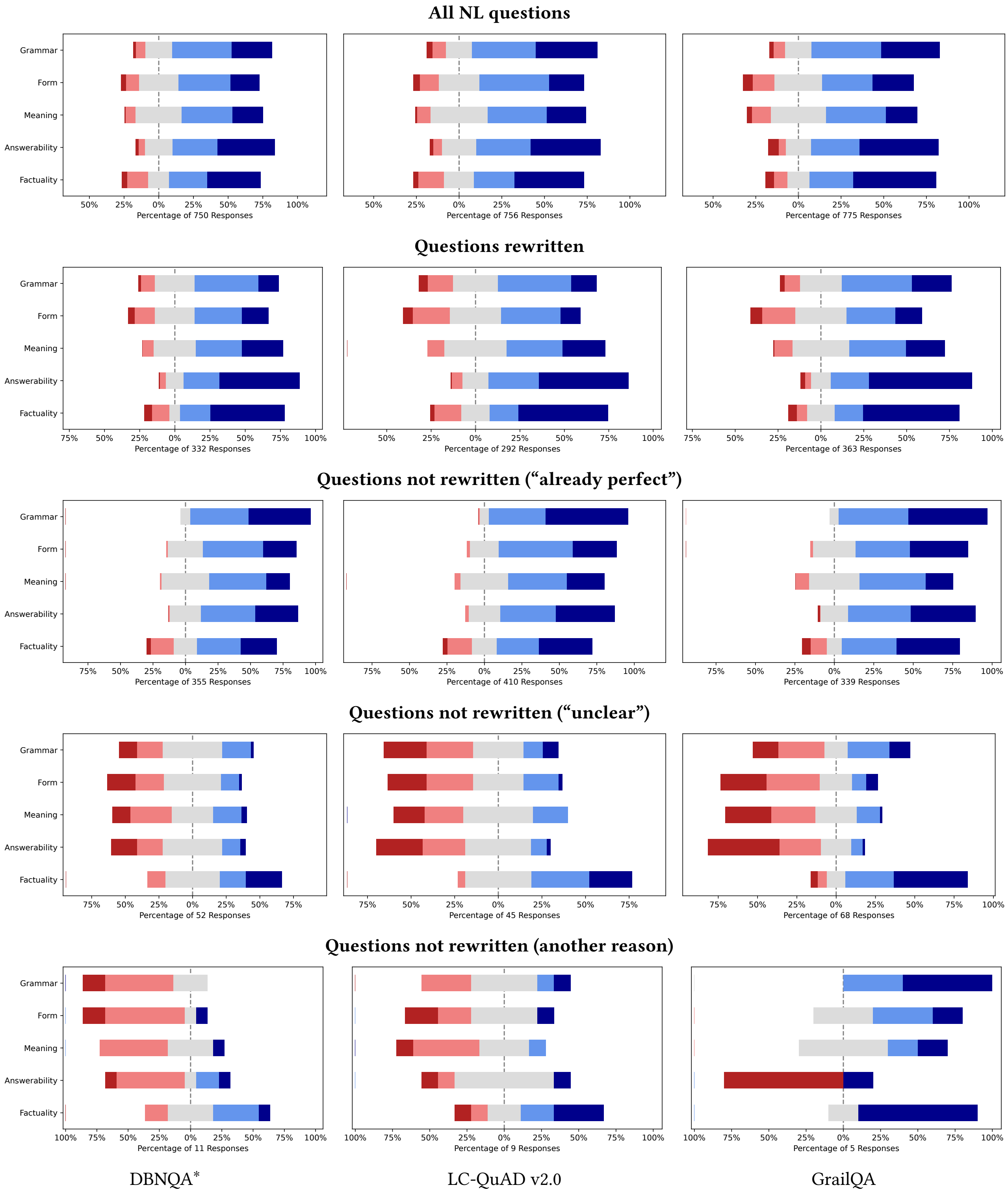

Figure 1: Likert scale rating results on all NL questions (Top), on questions which were rewritten (Second row), on questions which were not rewritten because the original question was "already perfect" (Middle), on questions which were "unclear" (Fourth row), and on questions which were not rewritten for another reason (Bottom), broken down per dataset (columns).

a short comment like "good" instead of a question, or else copy-pasting parts of the instructions into the rewrite field.

Whenever crowdsourced responses are excluded, additional responses are requested, so that every sampled NL question is annotated (and potentially rewritten) with acceptable responses by at least three different crowd workers.

The results of the Likert scale ratings are shown in Fig. 1. The scales are oriented so that the farther to the right the scale lies, the more natural the questions are considered by the crowd workers. The differences between the rows are intuitive since the rows group the responses in terms of the reason given for whether the original question has been rewritten or not. Specifically, the middle row reflects responses where the crowd worker deems the original question to be "already perfect" and hence abstains from rewriting the question. This is also the row with the highest ratings over all five Likert scales.

The bottom two rows also show a consistency between the Likert scale ratings and the reason given why the original question was not rewritten. However, here the ratings are mostly negative compared to the distribution over all responses. The bottom two rows' ratings are also negative compared to the second and middle rows from the top, which reflect the original question having been rewritten or being "already perfect." One interesting exception is that the Likert scale "factuality" is rated highly even in the bottom two rows of Fig. 1. It is possible that the distinction was not made clear to the crowd workers between whether a question indicates a very terse and factual answer or a longer, more descriptive one. Alternatively, it may be possible for a question to clearly indicate that its proper answer is factual, but that what is asked is so unclear that the question cannot be improved.

Overall, the consistency of ratings across the five Likert scales over the 750 sampled NL questions as rated in the approved crowdsourced responses is calculated as Cronbach's $\alpha = 0.707$, which is designated as "acceptable."

4.3 Task 2: Validate and Vote

We then use the rewritten questions from the previous task to establish which version of an NL question is the better formulation. For every original question where at least one rewrite was provided by crowd workers in Task 1, we take the original question and up to three rewrites, shuffle the order, and ask a different set of crowd workers to choose which version of the question is the best way of asking. See the GitHub repository for the specific instructions given to crowd workers.

For this task, since the response type is very simple and if less than three rewritten questions were generated there is always at least one non-option which the crowd worker technically can choose, quality control consists of removing responses from crowd workers who repeatedly choose non-options.

Each question and its rewrites are validated by at least three crowd workers. If the Task 2 result is a clear majority for any specific version of the question, then that is the question carried forward into the rewritten questions test collection. If there is not a clear majority given a set of original question and its rewrites, two more crowd worker validations are requested until a majority vote emerges. The distribution of responses is displayed in Fig. 2,

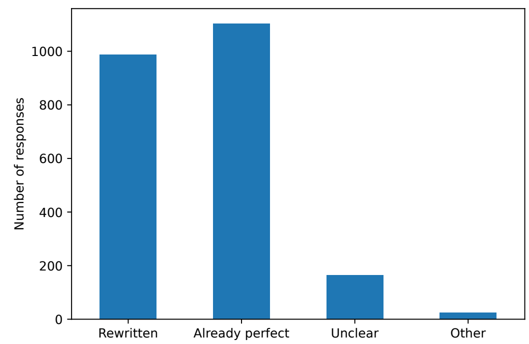

Figure 2: Histogram of frequencies over whether a rewritten question was provided ("rewritten") or otherwise reasons given for not rewriting (the original question was "already perfect" or "unclear," or else some "other" reason).

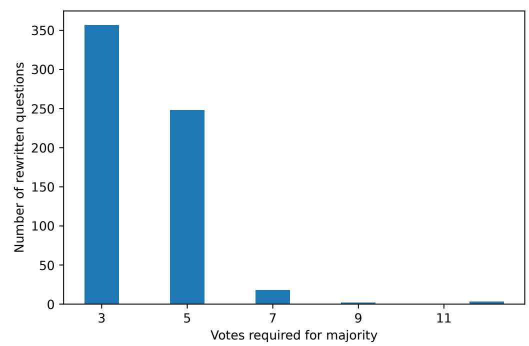

Figure 3: Histogram of frequencies over how many votes were required to determine a majority in favour of one version of the original or rewritten question.

while the number of crowd worker responses required to reach a majority is shown in Fig. 3. In total, 541 of the 750 questions are rewritten in the new collection; see Table 4 for a breakdown on specific subsets.

5 EXPERIMENTS

We compare model performance on the original versus rewritten NL questions in our samples. Specifically, we use neural KGQA models trained on the DBNQA* [16] and GrailQA [11] datasets.[1] The question we seek to answer is how quality improvements on the input NL questions impact the answer prediction effectiveness of the models.

5.1 Experimental Setup

For each of the KGQA datasets, the underlying KG is provided by a Virtuoso triplestore instance. For DBNQA*, DBpedia 2016 is the KG used to execute formal queries to retrieve answers. For GrailQA,

[1]Since we could not find papers with open source code addressing LC-QuAD v2.0 and there are still no models on the corresponding leaderboard, this dataset is not included in our experiments.

Freebase is served as the KG, following the instructions provided by Gu et al. [11].[2] This includes using their processed version of Freebase to make it fully compatible with the relevant Resource Description Framework (RDF) standard.

The GrailQA models rely on entity linking, which is provided for the full GrailQA validation split. In order to compare the original and rewritten question samples under equivalent conditions, the rewritten questions are identified with the original questions' query ID to apply the same entity linking to the rewritten NL question.

5.2 Methods

We use seven different neural KGQA methods to test the effect of rewritten NL questions on KGQA performance. These are sequence-to-sequence neural models with an encoder-decoder motif, where all but one are used to generate the formal query as a sequence of tokens. The exception is Ranking+BERT [11], where a neural model is used as a ranker to rank generated candidate formal queries.

Three methods are variations of the Neural Sparql Machine (NSpM) [22, 23, 33] architecture, including the NSpM baseline, NSpM+Att1, and NSpM+Att2 models. They are all based on Tensorflow NMT. NSpM+Att1 features a normed Bahdanau [2] attention mechanism, while NSpM+Att2 uses a scaled Luong [18] attention mechanism. All three NSpM models are specified with 2 layers and a dropout coefficient of 0.2. They are also all trained for 50,000 training steps.

Two methods, ConvS2S [10] and Transformer [29], are adapted from machine translation between natural languages to semantic parsing for KGQA. We rely on the sequence-to-sequence model implementations in Pytorch.[3] To better support the models, the both NL and formal query data is pre-processed with sub-word tokenization, specifically Byte Pair Encoding (BPE) [20] using Sentencepiece.[4] The other hyperparameters for ConvS2S and Transformer are kept as default, except notably the training data is not shuffled between epochs during training, and the models are trained in a case-sensitive manner.

The five models mentioned thus far are all trained on the training split of DBNQA*, which has been shuffled. The models predict formal queries from the NL questions in the full test split of DBNQA*, as well as the original sample of 250 NL questions from the test split, and the rewritten NL questions of the same sample.

Next, we use the methods using a pre-trained model based on BERT [8] provided by Gu et al. [11].[5] Specifically, the model is an LSTM-based sequence-to-sequence which uses uncased base-BERT for encoding, and is fine-tuned on GrailQA train split. The Transduction+BERT method uses this model for generating a formal query in an auto-regressive manner. In contrast, Ranking+BERT uses this model to rank candidate formal queries.

5.3 Results and Analysis

With the methods described above, we achieve the KGQA performance results listed in Table 5. We use two performance measures, Exact Match (EM) and F_1, to quantify effectiveness. Exact match compares the predicted formal query to the ground truth formal query. For DBNQA*, EM is 1.0 for an instance if and only if the two strings are identical. Meanwhile, using the provided evaluation script with GrailQA [11], the predicted and ground truth formal queries are both converted to query graphs and are considered as exactly matching if the graphs are isomorphic. The F_1 measure is based on the precision and recall of comparing answer sets. For the KGQA models evaluated on DBNQA*, if both the ground truth answer and the predicted answer are empty sets, the score for an instance is 1.0. This follows the example of Usbeck et al. [28].

In Table 5, we observe that the original questions in our sample (IQN-KGQA) may differ in terms of mean performance when compared to the full subset from which the sample was taken. The difference can be either either lower (e.g., Transformer) or higher (e.g., ConvS2S) on the sample than on the full subset. However, performance on original questions are of the same magnitude for both the full subset and sample across all methods. This holds true for both DBNQA*-test and GrailQA-dev.

In contrast, performance is reduced drastically when predicting on the rewritten questions. The three methods with the highest performance overall, Transformer, Ranking+BERT, and Transduction+BERT, all show that performance in both EM and F_1 is reduced by a large fraction (up to 78%) when predicting on the rewritten questions compared to predicting on the original questions. This trend is also followed by the NSpM+Att2 results, while the remaining models, NSpM baseline, NSpM+Att1, and ConvS2S all show some deviations. These models achieve a higher performance in one or both measures on the sampled original questions compared to the full DBNQA*-test split. Excepting the NSpM baseline, however, performance in both measures is less when predicting on the rewritten questions than the original questions.

We indicate statistical significance in Table 5 on the performance of the rewritten questions sample compared to the original questions sample for each KGQA model. A single dagger (†) indicates that the p-value was below $\alpha = 0.05$, while a double dagger (‡) indicates that the p-value was less than the Bonferroni-corrected threshold of $\frac{\alpha}{7}$ based on the seven comparisons made for each dependent variable.

6 DISCUSSION

The present work addresses data quality and collects improved formulations of NL questions, to yield the IQN-KGQA test collection. We reflect on the data collection process, discuss possible uses of our test collection, and identify limitations.

6.1 Data Collection

We have followed a similar procedure as the crowdsourced paraphrasing and cross-validation used in the construction of several large-scale KGQA datasets described in Sect. 2. Unlike the reported crowdsourcing of prior datasets, we have involved the crowd workers in a consideration of language quality and question naturalness, by soliciting ratings on the five unnaturalness dimensions, immediately prior to paraphrasing an original NL question. The Likert ratings themselves provide a perspective into how crowd workers see NL questions that should be rewritten compared to those that should not or cannot be rewritten.

[2] https://github.com/dki-lab/Freebase-Setup
[3] https://github.com/bentrevett/pytorch-seq2seq/
[4] https://github.com/google/sentencepiece
[5] https://github.com/dki-lab/GrailQA/

Table 5: Results on the DBNQA* and GrailQA datasets. The full subset refers to the original benchmarks and is included for reference. The IQN-KGQA dataset contains a sample of 250 questions per dataset. Performance is reported on the original questions in the sample as well as on their rewritten variant with improved naturalness. Significance is tested between the rewritten and original questions.

Dataset	Subset	Method	Full subset		IQN-KGQA			
			Original questions		Original questions		Rewritten questions	
			EM	F_1	EM	F_1	EM	F_1
DBNQA*	Test	NSpM baseline	0.000	0.013	0.000	0.020	0.004	0.016
		NSpM+Att1	0.081	0.119	0.085	0.105	0.033†	0.050†
		NSpM+Att2	0.089	0.132	0.081	0.117	0.028†	0.050‡
		ConvS2S	0.091	0.138	0.121	0.152	**0.036**‡	0.048‡
		Transformer	**0.177**	**0.260**	**0.166**	**0.254**	**0.036**‡	**0.067**‡
GrailQA	Dev	Ranking+BERT	**0.510**	**0.583**	**0.452**	**0.540**	**0.372**	**0.452**†
		Transduction+BERT	0.337	0.364	0.296	0.339	0.208†	0.251†

A majority of sampled NL questions were marked by some of the crowd workers as needing rewriting. Furthermore, during the second crowdsourced task, we see that the for most rewritten questions, the preferred version emerges quickly—in most cases with 3–5 votes. The resulting test collection has a majority (541 of 750) of its NL questions rewritten from their original form. This indicates that crowd workers have found room for improvement even after the initial paraphrasing and cross-validation undertaken in the original KGQA datasets' construction. This proportion of question rewrites also indicates that all three KGQA datasets can benefit in terms of question naturalness from extensive NL question rewriting.

6.2 Utilization

The present work describes a process of improving NL questions for KGQA datasets. This shows the value of additional rounds of rewriting and quality control when creating NL questions via crowdsourcing. However, the reduced performance of KGQA models on the sample with rewritten NL questions also calls into question the overall approach of relying heavily on crowdsourcing for large scale KGQA dataset construction.

We encourage other researchers to report their performance on our IQN-KGQA test collection as well as the test splits of the KGQA datasets on which they train their models. This will serve to keep the true KGQA performance in perspective. The reduced performance caused by rewritten NL questions illustrates that KGQA models are effectively overfitting on their datasets and do not generalize to natural question formulations. Our IQN-KGQA collection can be used to guard against this.

Our crowdsourcing designs can be utilized in future large-scale KGQA dataset construction efforts. The numerical ratings on the various dimensions of unnaturalness (in Task 1) may be used as quality control. Our collection could also be utilized for automatic question rewriting using, e.g., for fine-tuning large language models, to generate question paraphrases to contribute to the pool of options that crowd workers can vote on (as in Task 2).

6.3 Limitations

We tried to simplify quality control of crowdsourcing by having some heuristics about what constituted a reasonable effort of rewrites, but these filters were perhaps not sufficient. There are examples where the crowdsourced rewriting and validation seem to fail to improve the NL question that is rewritten. For example, the original question "Which past members of the labelle also sang somebody loves you baby (Blackstreet & Ma song)?" was voted down in favor of the rewritten question "What song is Patti LaBelle famous for ?"

Our test collection is of small scale, yet has been relatively expensive to produce, on the order of US$1000. Although crowdsourcing labor may be an economic way to scale up data annotation, there remains a question of how involved the manual quality control should be from the researchers' side.

7 CONCLUSION

We have investigated the dimensions of unnaturalness in the nominally natural language questions found in several modern large-scale knowledge graph question answering (KGQA) datasets. Specifically, we have developed a coding scheme to evaluate the naturalness of NL questions. We have also used crowdsourcing to rewrite such NL questions in KGQA datasets to be more genuinely natural. By combining language quality evaluation with NL question rewriting, we have attempted to prime crowd workers with attention towards language quality. From these rewritten NL questions, we have created the IQN-KGQA test collection with grounding in each of the three major knowledge graphs (KGs) addressed in previous KGQA research: DBpedia, Freebase, and Wikidata. This test collection can put KGQA performance in a more realistic perspective compared to testing KGQA systems on validation and test splits created with the exact same procedure as the training split. We have experimentally shown the impact of our test collection on the performance of KGQA models compared to performance on the corresponding sample of original NL questions and found that model performance deteriorated substantially when a more natural formulation of the same questions was provided. This suggests that existing models do not generalize well to genuinely natural questions. The present work represents an initial effort to better understand ways to improve the naturalness of NL questions for KGQA and to ensure that KGQA performance is evaluated with genuinely natural questions.

REFERENCES

[1] Jaime Arguello, Adam Ferguson, Emery Fine, Bhaskar Mitra, Hamed Zamani, and Fernando Diaz. 2021. Tip of the Tongue Known-Item Retrieval: A Case Study in Movie Identification. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval (CHIIR '21)*. 5–14.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:1409.0473 [cs.CL]

[3] Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-Based Question Answering with Knowledge Graph. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers (COLING '16)*. 2503–2514.

[4] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP '13)*. 1533–1544.

[5] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale Simple Question Answering with Memory Networks. arXiv:1506.02075 [cs.LG]

[6] Qingqing Cai and Alexander Yates. 2013. Large-scale Semantic Parsing via Schema Matching and Lexicon Extension. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL '13)*. 423–433.

[7] Nilesh Chakraborty, Denis Lukovnikov, Gaurav Maheshwari, Priyansh Trivedi, Jens Lehmann, and Asja Fischer. 2021. Introduction to Neural Network-based Question Answering over Knowledge Graphs. *WIREs Data Mining and Knowledge Discovery* 11, 3 (2021), e1389.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (NAACL-HLT '19)*. 4171–4186.

[9] Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia. In *Proceedings of the 18th International Semantic Web Conference (ISWC '19)*. 69–78.

[10] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17)*. 1243–1252.

[11] Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond I.I.D.: Three Levels of Generalization for Question Answering on Knowledge Bases. In *Proceedings of the Web Conference 2021 (WWW '21)*. 3477–3488.

[12] Ann-Kathrin Hartmann, Edgard Marx, and Tommaso Soru. 2018. Generating a Large Dataset for Neural Question Answering over the DBpedia Knowledge Base. In *Workshop on Linked Data Management, co-located with the W3C WEBBR 2018 (WEBBR '18)*.

[13] Ida Kathrine Hammeleff Jørgensen and Toine Bogers. 2020. "Kinda like The Sims... But with Ghosts?": A Qualitative Analysis of Video Game Re-Finding Requests on Reddit. In *International Conference on the Foundations of Digital Games (FDG '20)*.

[14] Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring Compositional Generalization: A Comprehensive Method on Realistic Data. In *Proceedings of the 2020 International Conference on Learning Representations (ICLR '20)*.

[15] Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A Survey on Complex Knowledge Base Question Answering: Methods, Challenges and Solutions. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI '21)*. 4483–4491.

[16] Trond Linjordet and Krisztian Balog. 2020. Sanitizing Synthetic Training Data Generation for Question Answering over Knowledge Graphs. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval (ICTIR '20)*. 121–128.

[17] Vanessa Lopez, Christina Unger, Philipp Cimiano, and Enrico Motta. 2013. Evaluating Question Answering over Linked Data. *Web Semant.: Science, Services and Agents on the World Wide Web* 21 (2013), 3–13.

[18] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. arXiv:1508.04025 [cs.CL]

[19] Rishiraj Saha Roy and Avishek Anand. 2021. Question Answering for the Curated Web: Tasks and Methods in QA over Knowledge Bases and Text Collections. *Synthesis Lectures on Inf. Concepts, Retr., and Services* 13, 4 (2021), 1–194.

[20] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural Machine Translation of Rare Words with Subword Units. arXiv:1508.07909 [cs.CL]

[21] Jiaxin Shi, Shulin Cao, Liangming Pan, Yutong Xiang, Lei Hou, Juanzi Li, Hanwang Zhang, and Bin He. 2020. KQA Pro: A Large Diagnostic Dataset for Complex Question Answering over Knowledge Base. arXiv:2007.03875 [cs.CL]

[22] Tommaso Soru, Edgard Marx, Diego Moussallem, Gustavo Publio, André Valdestilhas, Diego Esteves, and Ciro Baron Neto. 2017. SPARQL as a Foreign Language. (2017). arXiv:1708.07624 [cs.CL]

[23] Tommaso Soru, Edgard Marx, André Valdestilhas, Diego Esteves, Diego Moussallem, and Gustavo Publio. 2018. Neural Machine Translation for Query Construction and Composition. In *ICML Workshop on Neural Abstract Machines & Program Induction (NAMPI v2) (ICML '18)*.

[24] Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gür, Zenghui Yan, and Xifeng Yan. 2016. On Generating Characteristic-rich Question Sets for QA Evaluation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP '16)*. 562–572.

[25] Alon Talmor and Jonathan Berant. 2018. 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) (NAACL '18)*. 641–651.

[26] Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. LC-QuAD: A Corpus for Complex Question Answering over Knowledge Graphs. In *Proceedings of the 16th International Semantic Web Conference (ISWC) (ISWC '17)*. 210–218.

[27] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, Michael Röder, and Giulio Napolitano. 2017. 7th Open Challenge on Question Answering over Linked Data (QALD-7). In *Proceedings of Semantic Web Challenges - 4th SemWebEval Challenge at ESWC 2017 (ESWC '17, Vol. 769)*. 59–69.

[28] Ricardo Usbeck, Michael Röder, Michael Hoffmann, Felix Conrads, Jonathan Huthmann, Axel-Cyrille Ngonga-Ngomo, Christian Demmler, and Christina Unger. 2019. *Semantic Web* 10, 2 (2019), 293–304.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS '17)*. 6000–6010.

[30] Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a Semantic Parser Overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJNLP '15)*. 1332–1342.

[31] Peiyun Wu, Xiaowang Zhang, and Zhiyong Feng. 2019. A Survey of Question Answering over Knowledge Base. In *Knowledge Graph and Semantic Computing: Knowledge Computing and Language Understanding*. 86–97.

[32] Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The Value of Semantic Parse Labeling for Knowledge Base Question Answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (ACL '16)*. 201–206.

[33] Xiaoyu Yin, Dagmar Gromann, and Sebastian Rudolph. 2021. Neural Machine Translating from Natural Language to SPARQL. *Future Generation Computer Systems* 117 (2021), 510–519.