Quda: Natural Language Queries for Visual Data Analytics

Category: Theory/model
Paper Type: Theory/model

1. Expert Query Collection

- 36 data tables from 11 domains
- Interview with 20 experts
- 920 expert queries

2. Paraphrase Generation

- Crowd sourcing
- 20 paraphrases per expert query

3. Paraphrase Validation

- Crowd + Machine
- Validation results:

  a1. Which video is the most liked?
  a2. What video has the most likes?
  a3. Please name the most liked video.
  a4. This is the most like video
  a5. Which video has the highest amount of likes?
  a6. This is the most like video

Fig. 1. Overview of the entire data acquisition procedure consisting of three stages. In the first stage, we collect 920 queries by interviewing 20 data analysts. Second, we expand the corpus by collecting paraphrases using crowd intelligence. Third, we borrow both the crowd force and a machine learning algorithm to score and reject paraphrases with low quality.

Abstract—Visualization-oriented natural language interfaces (V-NLIs) have been explored and developed in recent years. One challenge faced by V-NLIs is in the formation of effective design decisions that usually requires a deep understanding of user queries. Learning-based approaches have shown potential in V-NLIs and reached state-of-the-art performance in various NLP tasks. However, because of the lack of sufficient training samples that cater to visual data analytics, cutting-edge techniques have rarely been employed to facilitate the development of V-NLIs. We present a new dataset, called Quda, to help V-NLIs understand free-form natural language. Our dataset contains 14,035 diverse user queries annotated with 10 low-level analytic tasks that assist in the deployment of state-of-the-art techniques for parsing complex human language. We achieve this goal by first gathering seed queries with data analysts who are target users of V-NLIs. Then we employ extensive crowd force for paraphrase generation and validation. We demonstrate the usefulness of Quda in building V-NLIs by creating a prototype that makes effective design decisions for free-form user queries. We also show that Quda can be beneficial for a wide range of applications in the visualization community by analyzing the design tasks described in academic publications.

Index Terms—Natural language, Dataset, Analytical tasks.

1 INTRODUCTION

Research in visualization-oriented natural language interface (V-NLI) has attracted increasing attention in the visualization community. Data analytics or datalow systems such as Articulate [47] and FlowSense [53] leverage V-NLIs to provide easy-to-use, convenient, and engaging user experience, and facilitate the flow of data analysis. Existing V-NLIs, however, face two key challenges [53]. On the one hand, making effective design decisions, e.g., choosing a proper visualization, is challenging because of the large design space. V-NLIs can address this issue by employing findings of empirical studies that provide design recommendations given analytic tasks and datasets [40]. However, addressing this issue requires V-NLIs to have a deep understanding of queries and datasets, e.g., distilling analytic tasks from queries before making a design choice.

On the other hand, due to the complex nature of human language, understanding free-form natural language input (or “query” in the following article) is non-trivial. Recent research in N-LI has focused mainly on rule-based language parsers to understand users intent [42, 46, 53]. Although effective, these approaches are narrow in usage scenarios because of limited grammar rules. Moreover, establishing rules is cumbersome and error-prone, and may require extensive knowledge in both visualization and natural language processing (NLP).

Learning-based approaches have reached state-of-the-art performance in various NLP tasks and have great potential in understanding free-form queries in V-NLIs. For example, text classification techniques categorize user queries into analytic tasks, which contributes to the choice of proper visualization. Large-scale training examples that cater to visual data analytics are critical to developing and benchmarking learning-based techniques to be deployed for V-NLIs. Techniques trained using such a corpus can also benefit a wide range of applications in the visualization community. Taking the above text classification technique as an example, it can be applied it to the analysis of design requirements in academic literature. Admittedly, the acquisition of such a corpus is difficult for two reasons. First, the target users of V-NLIs are often identified as data analysts who are knowledgeable in analytic tasks [46, 53]. Therefore, the high-quality corpus should reflect how expert users interact with data under various tasks. Second, compared to general-purpose datasets, such as existing corpora collected via social media [25] or online news corpora [53], expert queries for V-NLIs are not naturally occurring.

In this paper, we aim to present a corpus that helps V-NLIs categorize queries into analytic tasks by training and benchmarking cutting-edge NLP techniques. Our corpus consists of a number of queries annotated with an analytic task, and features high-quality, large-volume queries designed in the context of V-NLIs. To accomplish this goal, we employ both expert and crowd intelligence. First and

[1]https://freenli.github.io/quda/
most importantly, we design a lab study and collect seed queries by interviewing data analysts. Next, to diversify the ways of saying something, we employ the crowd to collect a number of sentential paraphrases for each query. These queries are restatements with approximately the same meaning. Finally, we design a validation procedure for quality control. The Quda dataset contains 920 utterances created by data analysts, and each are accompanied by 14 paraphrases on average. All queries are annotated with one of ten analytics tasks.

We present the characteristics of Quda in comparison to other popular datasets using statistical analysis and benchmarking a range of machine/deep learning models for text classification and paraphrase generation. Although these corpora vary in scale and diversity, our corpus is complementary to previous work because it targets at the domain of visual data analytics. Next, we demonstrate how our dataset can benefit V-NLIs by designing and developing a prototype that incorporates a well-known text classification model trained on our dataset. Owing to the ability of recognizing analytic tasks for free-form textual input, our prototype makes effective design decisions. Finally, we explore the potential use of Quda to support the analysis of design requirements described in academic papers. We compare task distribution in the sports and Urban Computing domains and find that analysts in Sports are more interested in correlation analysis while experts in Urban Computing highlight “Characterize Distribution”. To summarize, the primary contributions of this paper are as follows:

- By employing both the expert and crowd intelligence, we construct a large-scale and high-quality corpus, named Quda, that contains 14,035 queries in the domain of visual data analytics.
- Building upon existing research on V-NLIs and task taxonomies, we define the space of queries for V-NLIs and highlight the scope of our corpus from six dimensions.
- To present the usefulness of Quda, we construct a V-NLI prototype that makes effective design decisions for free-form queries. We also demonstrate that our corpus can benefit a wide range of applications in the visualization community, such as the analysis of design requirements in academic papers.

Our corpus is the first attempt to construct large-scale corpus for V-NLIs. With the release of Quda, we hope it will stimulate the publication of similar-size datasets for the visualization community and the emergence of increasing benchmark against which the state-of-the-art models can be trained and evaluated.

2 RELATED WORK

Our work draws on prior research in V-NLIs, datasets for visualization research, and similar corpora in the NLP domain.

2.1 NLI for Data Visualization

Our research is motivated by recent progress in V-NLIs, which provides engaging and effective user experience by combining direct manipulation and natural language as input.

Some research emphasizes on the multi-modal interface that interacts with users using natural language. Back to 2010, Articulate [47] identifies nine task features and leverages keyword-based classification methods to understand how user’s imprecise query weighted in each feature. Sisl [12] is a multi-modal interface that accepts various input modes, including point-and-click interface. NL input in textual form, and NL using speech. Analyza [14] is a system that combines V-NLI with a structured interface to enable effective data exploration. To address the ambiguity and complexity of natural language, DataTone [18] presents a mixed-initiative approach to manage ambiguity in the user query. Flowsensen [33] is a V-NLI designed for the dataflow visualization system. It applies semantic parsing to support natural language queries and allows users to manipulate multiple views in a dataflow visualization system. Several commercial tools, such as Microsoft Power BI [2], IBM Watson Analytics, Wolfram Alpha [6], Tableau [4], and ThoughtSpot [5], integrate V-NLIs to provide better analytic experience for novice users.

Another line of research targets the conversational nature of V-NLI. Fast et al. proposed Iris [15] that is a conversational user interface that helps users with data science tasks. Kumar et al. [23] aimed to develop a data analytics system to automatically generate visualizations using a full-fledged conversational interface. Eviza [42] and Evizeon [20] are visualization systems that enable natural language interactions to create and manipulate visualizations in a cycle of visual analytics. Similarly, Orko [49] is a prototype visualization system that combines both natural language interface and direct manipulation to assist visual exploration and analysis of graph data.

The aforementioned approaches are mainly implemented using rule-based language parsers, which provide limited support for free-form NLI input. Instead of generating visualizations, Text-to-Viz [13] aims to generate infographics from natural language statements with proportion-related statistics. The authors sampled 800 valid proportion-related statements and built a machine learning model to parse utterances. Although promising, Text-to-Viz does not support queries for a broad range of analytic activities.

The performance and usage scenario of V-NLIs highly depend on language parsers. The cutting-edge NLP models have reached close-to-human accuracies in various tasks, such as semantic parsing, text classification, paraphrase generation, etc. However, few have been applied to visualization-oriented V-NLIs. We argue that the release of the high-quality textual dataset would assist the training and evaluation of NLP models in the domain of data analytics and visualization.

2.2 Datasets for visualization research

An increasing number of studies in the visualization community have employed supervised learning approaches in various scenarios such as mark type classification [41], reverse engineering [37], color extraction, etc. The capability of these approaches, however, highly relies on the availability of massive datasets that are used for training and evaluation.

Some datasets consisting of visual diagrams are constructed and open-sourced for visualization research. For example, Savva et al. [41] compiled a dataset of over 2,500 chart images labeled by chart type. Similarly, the Massvis dataset [9] contains over 5,000 static chart images, and over 2,000 are labeled with chart type information. Beagle [5] embeds a Web Crawler extracting more than 41,000 SVG-based visualizations from the web, and all are labeled by visualization type. Viznet [28], which collects about 4.8 million images from scientific literature and classifies them into five categories, including equation, diagram, photo, plot, and table. Poco and Heer [37] compiled a chart corpus where each chart image is coupled with bounding boxes and transcribed text from the image. Instead of compiling a corpus containing chart images, Quda includes queries accompanied by analytic tasks for visualization-oriented V-NLIs.

Viznet, on the other hand, collects over 31 million real-world datasets that provides a common baseline for comparing visual designs. Though containing rich textual information, Viznet can hardly be applied to provide training samples for learning-based NLP models. On the contrary, Quda is designed for helping V-NLIs understand utterances describing analytic tasks in information visualization.

2.3 Corpora in NLP Domain

Our corpus is primarily designed for helping V-NLIs classify queries into analytic tasks. Therefore we first survey related corpora for text classification. Second, we introduce a paraphrase dataset because our corpus can support paraphrasing tasks to some extent. Third, we present prior corpora for text-to-SQL because they are in the domain of data analytics, which is similar to Quda.

Text classification is the task of categorizing a piece of textual content into an appropriate category. A number of voluminous datasets have been proposed. For example, AG’s News corpus [55] includes 496,835 news articles accompanied by 4 high-level categories. Sogou News corpus [55] consists of 90,000 training examples and 12,000 testing examples. Each of them are classified into 5 categories including “sports”, “finance”, “entertainment”, “automobile”, and “technology”. Similar corpora [55] are DBPedia, Yelp Review Polarity, Yelp Review Full, etc. Though large in scale, sentences in the aforementioned
corpora are different from V-NLI queries in syntax and semantics. Akin to Quda, TREC [49] is a corpus for question answering. It has two versions, i.e., TREC-6 and TREC-50. TREC-6 contains questions assigned with one of six labels, i.e., abbreviation, description and abstract concepts, entities, human beings, locations, and numeric values. Whereas TREC-50 has finer-grained categories for each question. Both have 5,425 training samples and 500 test samples. Quda is different from TREC in two aspects: 1) it is three times larger in scale, and 2) TREC falls in different domains compared to Quda. Given an original sentence, techniques for paraphrase generation aim to generate text that conveys the same meaning as the original one. Rich and voluminous paraphrase datasets have been constructed to support learning-based models for paraphrase generation. For example, Twitter URL Paraphrasing Corpus [25] is a continuously growing paraphrase dataset that links tweets through shared URLs. Quora Question Pairs [3] contains over 400,000 potential duplicate questions pairs. MSCOCO [31] contains 328,000 images associated with 5 different sentences describing each image. The key difference between Quda and the existing paraphrase corpora is the application domain, which affects lexical distribution and syntax. Our corpus is the first attempt to build a paraphrase corpus for data analytics. Paraphrase generation was an important task that is influential to downstream tasks such as dialog systems, semantic parsing, and information retrieval. We envision that the release of Quda could foster research in paraphrase generation and broad NLP tasks for data analytics.

Text-to-SQL is the task that requires a deep understanding of natural language and databases and mapping it to meaningful executable SQL queries. Example corpora are WikiSQL (80,654 queries) [50], Spider (10,181 queries) [54], Advising (3,898 queries) [17], Yelp and IMDB (259 queries) [52], MAS (196 queries) [29], and SCHOLAR (6,160 queries) [21] to name a few. These corpora are powerful in applications such as database-oriented visualization. Quda can be distinguished from the corpora mentioned above from two aspects. First, queries in text-to-SQL corpora are not labeled with analytic tasks, on which the choice of visualization highly relies [40]. The lack of task information would hinder V-NLI from choosing a proper visualization to respond to a query. On the contrary, our study focuses on constructing a corpus for 10 low-level analytic tasks. Second, Quda covers a broader range of analytic tasks compared to text-to-SQL corpora. As discussed in Spider [54], the existing corpora are limited in SQL operators, such as ORDER BY, GROUP BY, and WHERE. Hence, the support for complex tasks such as “Find Anomalies” or “Correlate” is not feasible for text-to-SQL corpora.

3 Problem Definition

Analytic tasks play a pivotal role in visualization design and evaluation. The understanding of tasks is beneficial for a wide range of applications in the visualization community such as visualization recommendation, computational linguistics for data analytics, etc. The goal of this paper is to propose a corpus in the domain of visual data analytics that facilitates the deployment of learning-based NLP techniques for V-NLIs. Specifically, our corpus focuses on queries that reflect how data analysts ask questions in V-NLIs. Due to the variation and ambiguity of natural language, the space of queries is intangibly large. To narrow down the scope of our corpus, we borrow ideas from prior work in visualization framework [7, 33, 39] and V-NLI for visualizations [46, 48] and identify Quda from six dimensions, i.e., abstraction level, composition level, perspective, type of data, type of task, and context-dependency.

Abstraction: Concrete. The abstraction level is one dimension of analytic tasks describing the concreteness of a query [39]. Abstract queries are generic and are useful for generalizing the concept behind a concrete task (an example is, “Find maximum”). These queries can be addressed in multiple ways based on the interpretation. However, to obtain a reasonable response from V-NLIs, an analyst should author a query that provides sufficient information [46], such as “Find top movies with most stars” and “Retrieve the country with the most population”. Therefore, our corpus focuses on the queries at the low abstraction level which expresses both tasks and values explicitly.

Composition: Low. The composition level describes the extent to which a query encompasses sub-queries [39]. Composition is a continuous scale from low-level to high-level. A query with a high composition level consists of multiple sub-queries and a V-NLI may answer it using multiple steps. For example, “For the movie with the most stars, find the distribution of salaries of the filming team.” A V-NLI first needs to identify the movie with the most stars. Then display the salary distribution of all staff in the filming team. As our corpus is the first attempt to collect queries for V-NLI, we focus on queries that are low in the composition level at this stage. We plan to include with high composition level in future research.

Perspective: Objectives. Queries in V-NLIs can be classified into two categories, i.e., objectives and actions [39]. Objectives are queries raised by analysts seeking to satisfy a curiosity or solve a problem. The actions in V-NLIs are executable steps towards achieving objects, and usually relate to interactive features of visualization artifacts, such as “Show the distribution using a bar chart,” or “Map the popularity of applications to x-axis.” However, to assume V-NLI users being knowledgeable in constructing effective visualization is virtually arbitrary. Instead of enumerating actions, we aim to collect queries that are objectives raised by data analysts.

Type of Data: Table. The type of dataset affects the syntax and semantics of queries. For example, analysts may seek to identify the shortest path between two nodes for a network dataset. However, such a query rarely occurs in the tabular dataset because links between items are not supported explicitly. Data and dataset can be categorized into five types [33], including table, networks & trees, fields, geometry, and clusters & sets & lists. At the date of paper submission, our corpus target at queries based on the tabular dataset, we argue that our corpus can support other types of data to some extent. For example, the task of “Find Nodes” in networks is similar to “Retrieve Value” in tabular data. We plan to extend our research to support other data types comprehensively in the future.

Type of Tasks: 10 Low-level Tasks. Collecting a corpus for all possible tasks is not feasible. Therefore, we turned to the related studies to identify analytic tasks on which to focus. In this study, we adopt the taxonomy proposed by Amar et al. [7] that categorized ten low-level analytic activities, such as Retrieve Value, Sort, etc. These tasks serve a good starting point for our research despite not being comprehensive.

Context Dependency: Independent. The conversational nature of V-NLI may result in queries that rely on contextual information [46, 48]. For example, a query “Find the best student in the class” may be followed by “Obtain the English score of the student.” The second query appears to be incomplete individually and is a follow-up query of the first query. Such a query is referred to as contextual queries, which is not the focus of our research. Our corpus focuses on queries that have complete references to tasks or values associated with a task.

4 Constructing Quda

Quda is the first part of an ambitious project. We aim to incorporate theories and techniques in the visualization and NLP domains to facilitate the process of data analytics. Thus far, we have collected 14,035 queries based on 36 data tables and 10 low-level analytic tasks. We plan to complete the construction of around 5 large-scale corpora in the next two years. In this work, we incorporate both expert and crowd intelligence in data acquisition, which can be divided into three stages. In the first stage, we conduct an interview study involving 20 data analysts to author queries based on 36 data tables and 10 low-level tasks. We derive 920 expert queries from the study results. For the second stage, we perform a large-scale crowd-sourced experiment to collect sentential paraphrases for each expert query, which are restatements of the original query with approximately the same meaning [50]. Finally, we design a validation stage involving both crowd force and a machine learning algorithm to ensure data quality. The third stage results in 13,115 paraphrases where each expert sentence is accompanied with 14 paraphrases on average. Figure 1 shows an overview of the entire data acquisition procedure.
5 Employing Expert Intelligence

In this section, we describe expert interviews that aim to collect professional queries from data analysts, given analytic tasks and data tables in different domains.

5.1 Participants and Apparatus

To understand how data analysts raise questions from data tables, we recruit 20 individuals, 10 females, and 10 males, with ages ranging from 24—31 years ($\mu = 25.45, \sigma = 2.80$). In our study, we identify "data analysts" as people who are experienced in data mining or visual analytics and have at least one publication in related fields. Most participants (16/20) are postgraduate students majoring in Computer Science or Statistics, and the rest are working as data specialists in IT companies. The experiments are conducted on a laptop (2.8GHz 4-Core Intel Core i7, 16 GB memory) on which participants read documents and create queries.

5.2 Tasks and Data Tables

Tasks play a vital role in authoring queries. Our study begins with the taxonomy of 10 low-level visualization tasks [4], including Retrieve Value, Compute Derived Value, Find Anomalies, Correlate, etc. The pilot study shows that participants may be confused about some tasks. For example, one commented, “Determine Range is to find the maximum and minimum values in one data field, which is similar to Find Extremum.” To help participants clarify the scope of each task, we compiled a document presenting each task from three aspects, i.e., general description, definition, and example sentences. The document is shown in the supplementary material.

Data tables provide a rich context for participants to create queries. Hence, to diversify the semantics and syntax of queries, we have prepared 36 real-world data tables covering 11 different domains, including health care, sports, entertainment, etc. Tables with insufficient data fields may not support some types of queries. For example, assume a table about basketball players has two columns, i.e., players’ name and their nationality. Participants may find it hard to author queries in the “Find Extremum” or “Correlate” categories which usually require numeric fields. Hence, instead of skipping the tasks, we allow participants to revise tables by adding new columns or editing existing ones if necessary. Moreover, we selectively choose data tables that have rich background information and explanation for each column so that participants can get familiar with them in a short time. All data were collected from Kaggle[5], Data World[6], and Google Dataset Search[7].

5.3 Methodology and Procedure

The interview began with a brief introduction to the purpose of the study and the rights of each participant. We then collected the demographic information of each participant, such as age, sex, experience in data mining and visual analytics. After that, participants were asked to familiarize themselves with 10 analytic tasks by reading the document.

In the training stage, participants were asked to author queries for 10 tasks based on a sample data table, which differs from those used in the main experiment. We instructed them to think aloud and resolved any difficulties they encountered. We encourage participants to author queries with diverse syntax structure and semantics.

The pilot results indicate that the participants were more enthusiastic in generating sentences with diverse syntax if they are interested in the context of the table. At the beginning of the main experiment, we presented 10 tables with a detailed explanation of the context and data fields to participants and encouraged them to choose two based on their interest. We randomized the presentation order of the 10 tasks. Participants were guided to author at least two queries given a table and a task, and no time limit was given to complete each query. To summarize, the target queries collected at this stage is $20(\text{participants}) \times 2(\text{tasks}) \times 10(\text{tasks}) \times 2(\text{queries}) = 800(\text{queries})$.

However, because some of the participants enjoyed the authoring experience and generated more sentences for some tasks, the total number of queries exceeded 800.

After the main experiment, a post-interview discussion was conducted to collect their feedbacks. Interviews were audio recorded and we took notes of the participants’ comments for further analysis. Each interview lasted approximately two hours, and we sent a $15 gift card to interviewees for their participation.

5.4 Results

The result of our interview study is a corpus containing 920 queries generated by data analysts. The length of queries ranged from 4 to 35, with an average of 11.9 words ($\sigma = 3.9$). We characterize the 920 queries from three aspects, and present how analytic tasks vary in these aspects. First, we derived basic statistics for each task, i.e., the number of sentences per task and the distribution of sentence length. Next, we counted the number of uni-grams (with stop-words removal) and bi-grams to measure the lexicon diversity (a larger number means the higher diversity). Finally, to evaluate the diversity at the sentence level, we compute the average pairwise BLEU score [14] for queries within a task (the lower, the higher diversity). We do not explain these metrics in detail, because they have been presented at length in the previous paper. Figure[15] demonstrates the distribution of each property.

We observe that “Sort” has the least number of sentences. Since we emphasize diversity during the interview, participants did not attempt to author more queries when they found it hard to create sentences with diverse structures. Therefore, we infer that participants found it hard to create diverse queries for “Sort”. This observation is confirmed quantitatively, whereas “Sort” has the least number of uni-grams and bi-grams compared to others. The average pairwise BLEU score is also high. Akin to “Sort”, “Characterize Distribution” has the highest pairwise BLEU score while the quantity of uni-gram and bi-gram is low. That is, queries are similar in the syntactic and sentential levels.

On the contrary, some tasks, such as “Find Anomalies”, “Correlate”, and “Filter” have lower BLEU score and more uni-grams & bi-grams, which means queries for these tasks are diverse in linguistics. We observed that participants found it enjoyable to write queries for these tasks. For example, one participant commented in the post-interview discussion, “Find Anomalies is an interesting task because it requires an in-depth understanding of the data table.” She further added, “I asked ‘Based on the relationship among rating, installs, and the number of reviews, which app’s rating does not follow the trend?’ because this kind of abnormal apps is worth investigating.”

5.5 Reflections and Discussion

The target corpus should meet a set of requirements as discussed in Section[3]. However, we did not attempt to overload participants by explaining all requirements during the study. Instead, we carefully guided them to avoid potential pitfalls in writing queries. In this section, we report the considerations and reflections.

How is the abstraction level concretized? Our corpus focuses on queries with low abstraction level. However, some queries are authored with a high abstraction level, which raises ambiguity in understanding the intent. For example, given a movie dataset, one participant wrote down “Sort all movies.” The phrase does not state clearly based on what criteria the movies should be sorted. Hence, to make queries more concrete, participants were asked to answer a question, “How do you draw a visualization to address this query?” The solution may reveal the implicit intention of the participants, and they were further suggested to revise the query by adding more details.

How do we lower the composition level? We sought to collect queries with a low composition level. However, identifying the extent to which the query encompasses sub-queries is difficult. For example, given a table about students’ performance in a class. One participant would like to know, “who is the best student in the class?” This query can be broken down as “Compute the total score of all subjects for each student” and “Find the student with the maximum score.” Thus, to identify an adequate composition level, we asked participants to bear in mind that, “Whether this query can be solved using one visualization at the first step?” Participants were suggested to revise the query until

https://www.kaggle.com/datasets
https://data.world/
https://datasetsearch.research.google.com/
the answer to the above question is “Yes”. Taking the above sentence as an example, one analyst might answer the query by using a stacked bar chart directly where the x-axis is the student ID while the scores of each subject are stacked at the y-axis. Hence, the composition level of the above query is appropriate.

**How should they focus on the objectives?** Some of our participants are visualization experts. Hence, they tend to raise queries such as “Show a histogram for . . . (one data field)”, for the task “Characterize Distribution”. As discussed in Section 3, we focused on objectives rather than actions. Therefore, we asked them to revise such queries by pretending that they have no prior knowledge of visualization.

**How to assign task labels?** Some queries belong to multiple tasks in nature. For example, the sentence “For applications developed in the US, which is the most popular one?” falls in both “Filter” and “Find Extremum” categories. The identification of all task labels is important to reflect the characteristics of each query. However, each query is attached to a single task label in our corpus based on participants’ intent. For example, if the above sentence is authored under the task “Find Extremum”, then it is then attached to the label. We plan to address this issue using two approaches in future research. First, we can post-process all queries using machine learning approaches to identify potential tasks they belong to. Second, we can improve the design of the interview study by asking participants to review their queries.

**How to diversify the syntax and semantics?** The diversity of the syntax and semantics is critical to ensuring the quality of our corpus. Instead of asking participants to use a different syntax, we instruct them to raise “interesting and meaningful” questions. By focusing on semantics, participants tend to construct queries with detailed information, which benefits the diversity at both lexicon and sentence levels. For example, one participant noted, “Retrieve Value is a kind of trivial task. However, it is non-trivial to me because I was thinking about which value is worth retrieving.” She further commented, “In the second table (containing demographics information in the USA), I asked ‘How many of the Orange County residents are Asian?’ because Asians are dominant in the Orange County. Such a question is meaningful, I believe.”

### 6 Borrowing Crowd Intelligence

Paraphrase acquisition is a common approach for data augmentation. The goal of this stage is to extend the corpus by collecting paraphrases based on 920 expert sentences using crowd intelligence. This stage can be divided into two steps, i.e., paraphrase generation and validation. All experiments were conducted on MTurk.

#### 6.1 Paraphrase Acquisition at Scale

In this step, we aim to acquire at least 20 paraphrases for each expert sentence. We followed the crowdsourcing pipeline [10] to collect paraphrases using crowd intelligence. We developed a web-based system that the crowd can access through MTurk. Our system records both crafted queries and user behavior, e.g., keystrokes and duration, for further analysis.

We describe our task as [38], “Rewrite the original text found below so that the rewritten version has the same meaning, but a completely different wording and phrasing” to help crowd workers understand our task. The interface also encourages the crowd to craft paraphrases that differ from the original one in terms of sentence structure. We demonstrate both valid and invalid examples to explain our requirements. The interface displays an experts’ sentence and instructs workers to rewrite it. After a sentence is submitted, workers are allowed to author paraphrases for other expert sentences, skip sentences, or quit the interface by clicking “Finish”. Finally, we use an open-ended question to collect workers’ comments and suggestions for this job. We sort 0.1$ as a bonus for each sentence after validation.

Based on prior work in corpus collection [10][29] and the pilot experiments, we established a set of rules in the main experiment to ensure the quality of the collected sentences. First, to collect paraphrases that are diverse in structure, we limited the maximum number of sentences one could create (20 in our study) to involve more crowd intelligence. Second, to obtain sentences with high-quality, workers were informed that the results would be reviewed and validated before they receive a bonus. Third, the interface recorded and analyzed their keystrokes and duration in crafting sentences to reject invalid inputs. For example, we rejected sentences that were crafted within 5 seconds or with less than 10 keystrokes. Fourth, to avoid duplicate sentences with other workers, we compared the input with all sentences in the database.

Our job has received positive feedback from the crowd. We were encouraged by comments like “This was fun, and it made me use my brain” and “This wasn’t easy, but I hope I did it right, according to your rules. Fun and challenging. : ) ”.

#### 6.2 Validation via Pairwise Comparison

The result of the first step is a corpus containing 18,400 crowd-generated paraphrases accompanied by 920 expert-generated sentences. The goal of this step is to filter out paraphrases that are semantically inequivalent compared to the expert sentence.

Due to the ambiguity in human language, asking workers directly to identify whether paraphrases are semantically equivalent to the expert sentence would be restrictive and unreliable. For example, given an expert’s sentence, “How many faculties are there at Harvard University?”, and a worker’s, “How many departments are there at Harvard?”, it is not clear whether the two sentences have the same meaning because “faculty” could mean both “department” and “professor”. Instead of asking whether a paraphrase is semantically equivalent, we asked comparison questions to capture the equivalence strength of one paraphrase with respect to others. To be specific, given an expert sentence, a worker is shown a pair of paraphrases written by the crowd and asked to choose which one has closer meaning to the expert’s sentence. Pairwise comparison is effective in capturing semantic relationships [35]. However, exhaustive comparisons, i.e., each paraphrase compares to all other paraphrases using a large number of workers, is unnecessary and cost-prohibitive. Hence, borrowing from the idea of relative attributes [35], we design a strategy to collect pairwise comparison dataset for all paraphrases, and train scoring functions to estimate the semantic equivalence scores of them.
for each expert sentence is \( \frac{m \cdot n}{2} \), where \( m \) is the number of paraphrases, and \( n \) is the number of pairwise comparisons per paraphrase. In our experiment, \( m = 20 \), and \( n = 5 \). To alleviate randomness, each comparison is completed by 5 unique workers, providing \( \frac{m \cdot n}{2} \cdot 5 = 250 \) comparison tasks for each expert sentence. Given 920 expert sentences, this produces a \( 250 \times 920 = 230,000 \) tasks for 6,304 unique workers.

We set up a web-based system to assist crowd workers in conducting comparison jobs. The system displays job descriptions, examples and requirements. Each job consists of 10 comparison tasks based on distinct expert sentences, along with two gold standard instances for quality control. We manually crafted a set of gold standard tasks that are unambiguous and have a clear answer. We rejected any jobs where one of the gold tasks was failed. To measure subtle differences in the semantics, we used a two-alternative forced-choice design for each comparison in which workers cannot answer “no difference”. The workers were paid $0.15 per job.

6.2.2 Estimating Semantic Equivalence Score

After collecting a set of pairwise comparison data, we employed relative attribute [35] to estimate the semantic equivalence score of each paraphrase candidate. Given an expert sentence \( e \), we had 20 training paraphrases \( \{ p_i \} \) represented by feature vectors \( \{ x_i \} \) in \( \mathbb{R}^d \). We have collected a set of ordered pairs of paraphrases \( O = \{(p_i, p_j)\} \) such that \( (p_i, p_j) \in O \iff p_i > p_j \), i.e., paraphrase \( i \) is more close to the expert sentence \( e \) in semantics than \( j \). Our goal was to learn a scoring function:

\[
s_e(x_i) = w^Tx_i
\]

such that the maximum number of the following constraints is satisfied:

\[
\forall (p_i, p_j) \in O : w^Tx_i > w^Tx_j
\]

To capture semantic features of each paraphrase, we employed Universal Sentence Encoder [11] and obtained a 512-dimension feature vector \( x_i \) for each paraphrase. We used a SVM-based solver [35] to approximate \( w^T \), and obtained the estimated score for 20 paraphrases using Equation 1.

We set a threshold \( T \), so that paraphrases with scores lower than \( T \) are filtered out. The scores range from \(-0.8 \) to \(1.05 \), and we set \( T = 0 \) to reject all paraphrases with negative scores. Finally, we obtained 13,115 high-quality paraphrases for 920 expert sentences.

7 QUDA CHARACTERISTICS

In this section, we demonstrate the characteristics of Quda as compared to other popular datasets through 1) statistical analysis of these corpora and 2) benchmarking of a range of learning-based approaches. Based on the potential usage of our corpus, we focus on two fundamental NLP tasks, namely, text classification and paraphrase generation.

7.1 Text Classification

7.1.1 Statistical Analysis

Quda contains 10 categories, 14,035 sentences with 12.62 average sentence length. Compare to some well-known datasets designed for benchmarking deep text classification models, i.e., TREC (5,952 sentences, six categories) and SST [45] (11,748 sentences, five categories), Quda not only has a comparable scale but offers a new sub-domain that focuses on query intention classification. These unique properties make Quda the largest corpus in the domain of visual data analytics to date.

7.1.2 Benchmarking

Assigning a task label to a query is critical in V-NLIs. Therefore, we conduct an experiment to evaluate how text classification models perform on our corpus, and reveal the characteristics of our corpus with different experiment settings.

Models: We conduct our experiment with four well-known text classification models as baselines. The first is a convolutional neural network CNN [22] built on top of pre-trained word vectors for sentence classification. The second model, C-LSTM [57], utilizes the strength of

![Fig. 3. The performance of four well-known text classification baseline models on Quda_expert and Quda_aug_expert in terms of F1 score.](image-url)

CNN and long short-term memory recurrent neural networks (LSTM). The third is the C-CNN model that adds a recurrent connection to the convolutional layer [43]. Lastly, we employ Ad-RNN [52], which uses adversarial and virtual adversarial training to improve the RNN model. We use the default parameter settings for all models.

Dataset: We have prepared two corpora in this experiment. Different data splits (train and test) may affect the performance of models. We split the dataset by experts. That is, 20 experts are split into train and test sets (Quda_expert). All queries created by the same experts and their paraphrases are in the same set. We observed that high-frequency words that appeared in the Quda queries may have strong correlations with the contents in the data tables, that is, these words or phrases are values in data tables or synonyms of column names. These correlations may introduce biases and make the model ignoring the real intentions of the queries, and therefore hurting the model generalization. Inspired by prior work [55], we further employ a data augmentation technique on Quda that replaces words or phrases with their synonyms. To be specific, we derive a list of 210 high-frequency words after stop-words removal. Then we manually categorize these words into five groups, i.e., Place, Person, Time, Date, and Others. Next, for each group, we compile a set of alternative words from WordNet [16], and replace high-frequency words with the alternatives randomly. Finally, we split the augmented corpus by experts and obtain Quda_aug_expert. To reduce bias in data split, we report the average F1 scores of 10-fold cross-validation to estimate the performance of models on these corpora.

Results: Figure 3 shows the performance of four baseline models on Quda corpora in terms of precision. Compare to the SOTA precision of the other datasets for text classification at a similar scale (e.g., TREC-98 90.07% [11]), the numbers in Figure 3 indicate that classifying the Quda queries is challenging, and this leaves room for developing new methods that specifically designed for this task. As shown in Figure 3, on average, the augmented corpus (i.e., Quda_aug_expert) increases the precision for about 3.14%. This proves that the data augmentation we introduced in this paper is useful for model generalization.

7.2 Paraphrase Generation

Paraphrase generation is important in NLP and can be beneficial for a wide range of downstream applications, such as question answering, semantic parsing, information retrieval, text summarization, among others [30]. Each expert query is accompanied by 14 paraphrases, and thus, our corpus can serve as a benchmark for paraphrase identification, extraction, generation, etc. We explore the properties of Quda in the context of paraphrase generation.

7.2.1 Statistical Analysis

Table 1 shows the statistical information of Quda in comparison with other paraphrasing corpora. The Quora dataset [3] is a collection of question paraphrases composed of over 148,130 pairs of duplicate questions. It has about 148,532 unique sentences. We follow the experimental protocols mentioned by Patro et al. [36] to sample 50K (Quora-50K) and 100K (Quora-100K) as training sets and use the rest of the 10K and 30K as test sets, respectively. The Twitter dataset [25], which is widely used for paraphrasing research, consists of a large amount of human-annotated sentence pairs, which are collected from Twitter by linking tweets through shared URLs. This dataset is growing
continuously and has 114,025 pairs of paraphrases (88,938 unique sentences) after a one-year collection (from 2016.10 to 2017.10) [45]. We sample 100K randomly as a training set (Twitter-100K) and 15K as the testing set. We construct paraphrasing datasets from Quda using two settings. First, we allow sentence repeat using permutations (Quda-78K). For example, given 15 sentences (1 expert sentence + 14 paraphrases) with the same meaning, we construct \(\binom{15}{2} = 105\) pairs. As a result, each sentence will appear in multiple paraphrase pairs, and we have 78,939 pairs consisting of 11,102 unique sentences. Second, we avoid sentence repeat by selecting 7 pairs randomly from 15 sentences, which results in 5,365 pairs (10,730 unique sentences) in total (Quda-5K).

For the Quda-5K and Quda-78K datasets, we split the train (90%) and test (10%) randomly as shown in Table 1.

Compared to the dataset like Quora or Twitter, Quda differs from them for both vocabulary space and design principles. Quda-5K and Quda-78K are tailored for data analytics, and the queries in Quda reflect how analysts may ask questions on the data. On the other hand, utterances in Quora or Twitter datasets are mainly for general purposes and therefore cover a large range of casual topics. In summary, although Quda-5K and Quda-78K are not competitive to Quora or Twitter datasets in terms of the number of pairs and unique sentences, it provides a new data source with new problems that complementarily to all existing paraphrasing corpora.

### 7.2.2 Benchmarking

We first define the task of paraphrase generation, then introduce the setup of the experiment. Finally, we present the benchmarking results to provide deep insights into our corpus in comparison to others.

**Task Definition:** We follow the description in Patro et al.’s work [36] to introduce the task of paraphrase generation. Given an original sentence and a reference sentence \((x', x)\) with almost the same meaning, we need to generate another sentence \(y\) so that the distance between \(y\) and the reference \(x'\) is minimal. The training data \(S\) consist of \(M\) pairs of paraphrases \(S = \{(x_i', x_i)\}_{i=1}^M\), where \(x_i\) and \(x_i'\) are paraphrases of each other.

**Models:** We choose three representative techniques for paraphrase generation. The first model (EDLPS) proposes a pairwise discriminator based encoder-decoder for paraphrase generation. Encoder, decoder, and discriminator are built upon LSTM [36]. Second, we choose VAE-SVG, which combines deep generative models (VAE) with sequence-to-sequence models (LSTM). This model can generate multiple paraphrases for a given sentence [19]. The third model (DiPS) can generate paraphrases similar in semantics yet diverse in structure. The model is based on a basic Seq2Seq framework and features in monotone sub-modal function maximization [29]. We use the same parameter setting on all datasets for each model.

**Evaluation Metrics:** The evaluation metrics include METEOR [27], BLEU [33], and TER [44]. Though these metrics were used originally for machine translation, prior research has shown that they correlate with human judgments in terms of evaluating the quality of the generated paraphrases [51]. BLEU score considers the modified n-gram precisions and poses penalty for paraphrases that are too short or too long. We use the unigram precisions to present the results. The higher the BLEU score is, the better the generated paraphrases are in quality. METEOR calculates word mapping, unigram precision and recall between the generated paraphrase and the reference sentence. A higher METEOR score reflects better paraphrasing quality. TER stands for Translation Edit Rate, which measures how much effort humans would have to perform to change a generated paraphrase to the reference. Lower TER scores mean better quality.

**Results:** Table 2 shows the performance of three models using four datasets under three metrics. Generally, models do not perform well on Quda-5K compared to other datasets. All models result in lower BLEU score, lower METEOR score, and high TER for Quda-5K and Quda-78K. Given the same experimental settings, the performance of the models highly relates to the volume of training sets. An interesting observation is that models perform better on Quora-50K than Quda-78K, albeit Quora-50K is larger in scale than Quda-78K. That is probably because the number of unique sentences in Quda-78K is six times smaller than those in Quora-50K.

### 8 Quda Applications

We explore the use of Quda in two applications, i.e., the natural language interface and the analysis of design tasks in academic literature in two domains.

#### 8.1 Natural Language Interface

Our research is motivated by the rising trend of V-NLI. With the large design space [53], one challenge faced by V-NLI design is choosing a proper visualization to answer an analytic question. The choice of visualization design relies highly on analytic tasks. Given the type of task, V-NLI can make an effective design choice based on a large body of empirical research [40]. In this section, we present the design and implementation of a V-NLI prototype, called FreeNLI that benefits from the construction of Quda. The features of FreeNLI have two aspects: (1) It accepts free-form natural language as input, and (2) forms effective design decision by recognizing analytic tasks from the input query.

#### 8.1.1 The FreeNLI System

Figure 4 shows the system architecture of FreeNLI consisting of three major components: a language parser that processes NL query, a pool of design rules that identify effective design decisions, and an interface that interacts with data analysts.

**Language Parser.** Given an NL query, FreeNLI parses from two aspects. First, it applies a well-trained text classification model to categorize the type of analytic tasks involved in the query. We use the Ad-RNN model trained on Quora_aug_expert, which has the best performance as described in Section 7.1. By default, the model classifies queries into one of 10 analytic tasks. However, analysts may ask queries beyond the scope. To recognize the exception, we categorize a query into “Others” if the maximum estimation confidence is lower than a threshold, which is set to 0.3 in the current prototype.
Second, the language parser identifies data fields for visualization. The current prototype uses simple string matching to determine which columns are involved in the visualization. Therefore, users are required to mention column names in the query. Due to the sheer variety of natural language expressions, data analysts may use an abbreviation or alternative terms to refer to data fields. For example, a data table on the performance of basketball players may have two columns, i.e., name and score. A data analyst may search for, "which player (name) has the best performance (score)?" The mapping from NL to data fields is challenging, and in future research, we plan to employ more advanced techniques to alleviate this restriction.

Design Decisions. With the data fields and the analytic task, FreeNLI offers effective design decisions based on the findings of an empirical study [40]. To be specific, we derive design decisions from three aspects, i.e., data transformation, design choice, and design ranking. First, to accommodate analytic tasks such as 'Tell me the distribution of movie’s budget'; we employ simple data transformation and aggregation techniques, such as binning, counting, sorting, deriving maximum, minimum, and average. Next, referring to the empirical study, FreeNLI has limited support for the type of visualizations (scatterplots, bar charts, and line charts), the number of data fields (one or two), and type of data fields (numerical and nominal). Finally, to help a user complete analytic tasks with efficiency, we choose and rank visualizations to minimize the completion time [40].

Interface. Figure 5(a) shows FreeNLI that provides a concise interface for a user to interact with. As a start, a user may select or upload a dataset. Then, the dataset is shown as a table with which the user is allowed to navigate raw data and raise hypotheses. Simple manipulations, such as sorting and filtering, are supported. The user can then raise queries for the data table. When a query is submitted, a pop-up window with effective visualizations will be displayed to answer the query. The user can hover over visual elements to explore detailed information using a tooltip. Since FreeNLI was for demonstration purposes, we kept the interface simple and provided limited functionality and customization in the current prototype. We plan to refine and augment FreeNLI iteratively from two directions. First, we plan to provide rich configuration for the visualization and data transformation, such as adjusting the size of binning, including pivoting, sampling, etc. Second, we will improve user experience with more features, such as exploratory analysis and history, etc.

8.1.2 Usage Scenario

We describe a usage scenario to evaluate the usefulness of FreeNLI. Suppose Jane is a fan of Hollywood movies, and she exhibits considerable interest in analyzing a movie dataset comprising nine fields, including budget, title, genre, votes, etc. The dataset contains a total of 2,681 records, and she aims to gain a comprehensive sense of the dataset and identify trends, distribution, or other insights worth sharing. After loading the movie dataset to FreeNLI, a data table is shown to allow Jane to explore the dataset by sorting and filtering. "I do not want to use the table to explore such a large dataset," Jane murmurs. Instead, she decides to ask questions to visualize the dataset. For an overview of worldwide gross profit, Jane types the query "How about the worldwide gross profit of different genres?". A bar chart pops up to display the distribution of Genre on Worldwide Gross profit (Figure 5(b)). She observes that adventure movies have the grossest profit compared to other types of movies. Jane wonders whether adventure movies are released more than others. She then asks "Show me the distribution of genre." FreeNLI shows a pie chart to address this query (Figure 5(d)).

Surprisingly, Adventure only accounts for 9.2% among all types of movies. Drama, Comedy and Action have more market share compared to Adventure. Jane infers that adventure movies are usually high in the budget, and only filming teams with enough resources can afford it. To verify the hypothesis, she asks "What is the correlation between Genre and Budget?". A bar chart is displayed showing that Adventure requires the most budget (71 million dollars on average) among all categories (Figure 5(c)). Finally, she concludes, "Though adventure movies have high worldwide gross profit, only limited filming teams with a high budget can afford it."

8.1.3 Discussions and Future Directions

A large body of empirical research has evaluated the effectiveness of various design decisions given analytic tasks and datasets. However, data analysts without visualization background may not benefit from these studies. FreeNLI recognizes analytic tasks for free-form natural language queries and helps data analysts gain insights into data with effective visual representations. However, despite its promise, the current prototype is limited in two aspects. First, input queries should articulate which data fields are involved. Although this step is common for many existing V-NLIs, it is not natural for free-form NL queries. Hence, we plan to remove this restriction by employing models in text-to-SQL to map NL queries to data fields. Second, the current prototype is not scalable because of the need to construct a large number of rule-based design strategies. The second issue can be resolved by building end-to-end models that can learn expert behaviors in authoring visualizations, which is a promising research direction.

8.2 Comparison of Two Application Fields

To demonstrate the usefulness of our corpus beyond V-NLIs, we analyze and compare the design requirements of two application domains, i.e., visual analytics of sports data and urban computing. To answer questions such as, "How tasks distributed in the two domains" and "What are dominant tasks in each domain", we first collect 54
publications from leading venues. Then, we manually gather task utterances from the publications. Next, we employ the Ad-RNN model trained on Quda-aug-expert and classify task utterances into 10 categories. We include an “Others” category to handle utterances that do not belong to any of the 10 tasks. Finally, we report the analytic results in comparing the two domains.

8.2.1 Data Collection
To ensure that papers are of high quality, we focus on those published in top venues in visual analytics, transportation, and sports analytics, such as IEEE transactions on visualization and computer graphics, IEEE Transactions on Intelligent Transportation Systems, Proceedings of the CHI conference, etc. In addition, to obtain design requirements, we selectively choose papers that 1) target at domain problems and 2) solve domain problems using visual analytics. Finally, we have collected 54 papers where 28 are in the urban domain and 26 are in the sports domain. The full paper list can be found in the supplementary material.

The text classification model trained on Quda may have better performance if the data for prediction are in the same vocabulary and syntax space as training samples. Our corpus features queries that are objectives, concrete in the abstraction level, and low in the composition level. Therefore, we focus on task utterances that share similar characteristics as our corpus. To gather text as required, we have gone through sections related to design tasks, requirement analysis, system design, and case study, and collect all possible utterances in the publications. Next, we manually filter out utterances that do not meet our requirements. As a result, we obtain 629 utterances in total, where 249 are from the sports domain and the rest fall in the urban domain.

8.2.2 Results
The task distribution of each domain is shown in Figure 6. The x-axis presents task categories, while the y-axis means the proportion of tasks in each field. We observe that some tasks are dominant in both domains, such as Cluster, and Correlation. Since the number of entities, e.g., candidate solutions or tactics, is large in both domains. Clustering is usually applied to group similar entities and provide an overview of the entire dataset. For example, in a billboard placement project in the urban domain, analysts ask, “How many groups of candidate solutions exist?” Similarly, analysts in the sports domain often seek to understand “What are the frequent patterns of tactics?” In the sports domain, Correlation ranks first among all tasks. By scrutinizing the paper, we find that analysts often need to understand the potential effect of a particular adjustment in tactics. They may ask questions such as, “How does a tactical adjustment influence the strokes and tactics”, and “What is the effect of a formation change?” An interesting finding is that publications in the urban domain constitute a larger portion of Characterize Distribution tasks (17.63%) compared to the sports domain (8.03%). Since the Urban domain usually involves the analysis of spatial-temporal data. Analysts may ask questions such as “How are the target trajectories distributed across the city”, and “What is the difference between weekday and weekend?”

9 DISCUSSIONS
Quda, the largest textual corpus in visual data analytics to date, is the first attempt to construct a large-scale corpus for V-NLIs. Though we have shown the usefulness of Quda in various application scenarios, some limitations exist in the current version of the corpus. First, according to the six dimensions discussed in Section 4, our corpus covers a limited scope of queries. For example, Quda does not include contextual queries that frequently appear in V-NLIs and systems with multimodal interaction [48]. Corpora are fundamental in machine learning research, and those catered for visual data analytics facilitate the development of machine/deep learning approaches in the visualization community. We plan to construct more textual corpora in the context of V-NLIs in future research.

Second, data tables have a strong influence on queries. Data tables with various domains would diversify queries in semantics and syntax. Our corpus is constructed with 36 data tables in 11 domains. Though enumerating all data tables is infeasible, we should collect queries based on the most representative ones and cover as many domains as possible. In future research, we will investigate the datasets used in the domain of visual data analytics.

Third, paraphrase validation ensures the quality of crowd-authored queries. However, our approach requires extensive crowd force for pairwise comparisons, which would not be scalable if the number of paraphrases increases. As Quda is a large-scale corpus with human validation, we plan to train a machine learning model on our corpus to distinguish paraphrases with low quality. Then the scalability issue can be alleviated.

10 CONCLUSION AND FUTURE WORK
In this work, we present a large-scale corpus, named Quda, to help V-NLIs understand natural language queries by training and benchmarking machine/deep learning techniques. Besides applications such as V-NLIs, our corpus is fundamental in various domains, including visualization recommendation, automatic visualization generation, etc. Quda contains 14,035 queries labeled with 10 low-level analytic tasks. To construct Quda, we first collect 920 expert queries by recruiting 20 data analytics. Then, we employ crowd force to author 20 paraphrases for each expert query. Next, to ensure quality for paraphrases, we borrow the crowd to collect the pairwise comparison dataset and compute semantic equivalence scores using relative attribute [35]. Finally, paraphrases with low scores are filtered out. We present four experiments to illustrate the usefulness and significance of Quda in different usage scenarios.

In the future, we plan to explore more research opportunities using Quda. First, multimodal interaction for visualization systems can improve user experience and system usability [40]. We have built FreeNLI as the first step to explore natural language as an input modality. We plan to continuously expand the scope of Quda and augment the prototype to support more usage scenarios such as contextual queries. Second, we plan to improve the process of corpus construction to balance the data quality and cost. The cost for expert query collection and paraphrase generation is reasonable. However, the process of paraphrase validation is expensive, and the cost will increase quadratically with the number of paraphrases. With the help of Quda, we plan to train a classification model to investigate scalable approaches for quality control.
References

[1] IBM Watson Analytics, Accessed February 3, 2020. https://www.ibm.com/watson-analytics
[2] Microsoft Power BI, Accessed February 3, 2020. https://powerbi.microsoft.com
[3] Quora Question Pairs, Accessed February 3, 2020. https://www.kaggle.com/c/quora-question-pairs
[4] Tableau Software, Accessed February 3, 2020. http://www.tableausoftware.com
[5] Thoughtspot, Accessed February 3, 2020. http://www.thoughtspot.com/
[6] Wolfram Alpha, Accessed February 3, 2020. http://www.wolframalpha.com
[7] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In IEEE Symposium on Information Visualization, 2005. INFOVIS 2005., pp. 111–117. IEEE, Oct 2005. doi: 10.1109/INFOVIS.2005.1532136
[8] L. Battle, P. Duan, Z. Miranda, D. Mukusheva, R. Chang, and M. Stonebraker. Beagle: Automated extraction and interpretation of visualizations from the web. In Proceedings of the CHI Conference on Human Factors in Computing Systems, CHI 18. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3174668
[9] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister. What makes a visualization memorable? IEEE Transactions on Visualization and Computer Graphics, 19(12):23062315, Dec. 2013. doi: 10.1109/TVCG.2013.234
[10] S. Burrows, M. Potthast, and B. Stein. Paraphrase acquisition via crowdsourcing and machine learning. ACM Trans. Intell. Syst. Technol., 4(3):1–21, 7 2013. doi: 10.1145/2483669.2483676
[11] D. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limitoci, R. S. John, N. Constant, M. Gujardo-Cepedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil. Universal sentence encoder, 2018.
[12] K. Cox, R. E. Grinter, S. L. Hibino, L. J. Jagadeesan, and D. Mantilla. A multi-modal natural language interface to an information visualization environment. International Journal of Speech Technology, 4(3/4):297–314, 07 2001. doi: 10.1023/A:1011669264972
[13] W. Cui, X. Zhang, Y. Wang; H. Huang, B. Chen, L. Fang, H. Zhang, J.-G. Lou, and D. Zhang. Text-to-viz: Automatic generation of infographics from proportion-related natural language statements. IEEE Transactions on Visualization and Computer Graphics, 26(1):906–916, Jan 2020. doi: 10.1109/TVCG.2019.2934785
[14] K. Dhamdhere, K. S. McCurley, R. Nahmias, M. Sundararajan, and Q. Yan. Analyza: Exploring data with conversation. In Proceedings of the 22nd International Conference on Intelligent User Interfaces, IUI 17, p. 493504. Association for Computing Machinery, New York, NY, USA, mar 2017. doi: 10.1145/3025171.3025227
[15] E. Fast, B. Chen, J. Mendelsohn, J. Bassen, and M. S. Bernstein. Iris: an conversational agent for complex tasks. In Proceedings of the CHI Conference on Human Factors in Computing Systems, CHI 18. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3174047
[16] C. Fellbaum. Wordnet : an electronic lexical database. Language, 76(3):706, 2000.
[17] C. Finegan-Dollak, J. K. Kummerfelder, L. Zhang, K. Ramathan, S. Sadasivam, R. Zhang, and D. Radev. Improving text-to-SQL evaluation methodology. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, July 2018. doi: 10.18653/v1/p18-1033
[18] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In Proceedings of the Annual Symposium on User Interface Software and Technology, UIST 15, p. 489500. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2807442.2807478
[19] A. Gupta, A. Agarwal, P. Singh, and P. Rai. A deep generative framework for paraphrase generation. In AAAI Conference on Artificial Intelligence, 2018.
[20] E. Hoque, V. Setlur, M. Topy, and I. Dykeman. Applying pragmatics principles for interaction with visual analytics. IEEE Transactions on Visualization and Computer Graphics, 24(1):309–318, Jan 2018. doi: 10.1109/TVCG.2017.2744684
[21] S. Iyer, I. Konstas, A. Cheung, J. Krishnamurthy, and L. Zettlemoyer. Learning a neural semantic parser from user feedback, 2017.
[22] Y. Kim. Convolutional neural networks for sentence classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar, Oct. 2014. doi: 10.1117/14.1181
[23] A. Kumar, J. Aurisano, B. D. Engieno, A. Johnson, A. Gonzalez, and J. Leigh. Towards a dialogue system that supports rich visualizations of data. In Proceedings of the Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 304–309. Association for Computational Linguistics, Los Angeles, Sept. 2016. doi: 10.18653/v1/W16-3639
[24] A. Kumar, S. Bhattachamshra, M. Bhandari, and P. Talukdar. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, pp. 3609–3619. Association for Computational Linguistics, Minneapolis, Minnesota, June 2019. doi: 10.18653/v1/N19-1363
[25] W. Lan, S. Qiu, H. He, and W. Xu. A continuously growing dataset of sentential paraphrases. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1224–1234. Association for Computational Linguistics, Copenhagen, Denmark. Sept. 2017. doi: 10.18653/v1/D17-1126
[26] W. S. Lasecki, E. Kumar, and D. Bobus. Conversations in the crowd: Collecting data for task-oriented dialog learning. In First AAAI Conference on Human Computation and Crowdsourcing, 2013.
[27] A. Lavie and A. Agarwal. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In Proceedings of the Second Workshop on Statistical Machine Translation, pp. 228–231. Association for Computational Linguistics, Prague, Czech Republic, June 2007.
[28] P.-S. Lee, J. D. West, and B. Howe. Vizimetrics: Analyzing visual information in the scientific literature. IEEE Transactions on Big Data, 4(1):117–129, March 2018. doi: 10.1109/TBDA.2017.2689038
[29] F. Li and H. V. Jagadish. Constructing an interactive natural language interface for relational databases. Proc. VILDB Endow., 8(1):7384, Sept. 2014. doi: 10.14778/2735461.2735468
[30] Z. Li, X. Jiang, L. Shang, and H. Li. Paraphrase generation with deep reinforcement learning. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3865–3878. Association for Computational Linguistics, Brussels, Belgium, Oct.-Nov. 2018. doi: 10.18653/v1/D18-1421
[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds., Computer Vision – ECCV 2014, pp. 740–755. Springer International Publishing, Cham, 2014.
[32] T. Miyato, A. M. Dai, and I. Goodfellow. Adversarial training methods for semi-supervised text classification, 2016.
[33] T. Munzner. Visualization analysis and design. CRC press, 2014.
[34] B. N. Patro, V. K. Kurmi, S. Kumar, and V. P. Namboodiri. Learning semantic sentence embeddings using sequential pair-wise discriminator, 2018.
[35] J. Poco and J. Heer. Reverse-engineering visualizations: Recovering visual encodings from chart images. Comput. Graph. Forum, 36(3):35363, June 2017. doi: 10.1111/cgf.13193
[36] M. Potthast, B. Stein, A. Barrón-Cedeño, and P. Rosso. An evaluation framework for plagiarism detection. In Proceedings of the International Conference on Computational Linguistics: Posters, COLING 10, p. 311318. Association for Computational Linguistics, USA, 2010. doi: 10.3115/v1/D10-1181
[37] A. Rind, W. Aigner, M. Wagner, S. Miksch, and T. Lammarsch. Task cube: A methodological framework for plagiarism detection. In First AAAI Conference on Artificial Intelligence, 2010.
[38] M. S. Bernstein. Principles for interaction with visual analytics. In Visualization and Computer Graphics, 24(7):2505–2512, July 2019. doi: 10.1109/TVCG.2018.2829750
[41] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer. Revision: Automated classification, analysis and redesign of chart images. In *Proceedings of the Annual Symposium on User Interface Software and Technology*, UIST 11, p. 393-402. Association for Computing Machinery, New York, NY, USA, 2011. doi: 10.1145/2047196.2047247

[42] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang. Eviza: A natural language interface for visual analysis. In *Proceedings of the Annual Symposium on User Interface Software and Technology*, UIST 16, p. 365-377. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2984511.2984588

[43] J. Shin, Y. Kim, S. Yoon, and K. Jung. Contextual-cnn: A novel architecture capturing unified meaning for sentence classification. In *IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 491–494, 2018.

[44] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhou. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pp. 223–231, 2006.

[45] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642. Association for Computational Linguistics, Seattle, Washington, USA, Oct. 2013.

[46] A. Srinivasan and J. Stasko. Orko: Facilitating multimodal interaction for visual exploration and analysis of networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):511–521, Jan 2018. doi: 10.1109/TVCG.2017.2745219

[47] Y. Sun, J. Leigh, A. Johnson, and S. Lee. Articulate: A semi-automated model for translating natural language queries into meaningful visualizations. In *Proceedings of the 10th International Conference on Smart Graphics*, SG10, p. 184-195. Springer-Verlag, Berlin, Heidelberg, 2010.

[48] M. Tory and V. Setlur. Do what i mean, not what i say? design considerations for supporting intent and context in analytical conversation. *IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2019.

[49] E. M. Voorhees and D. M. Tice. The treec-8 question answering track: Evaluation. In *TREC*, vol. 1999, p. 82. Citeseer, 1999.

[50] S. Wang, R. Gupta, N. Chang, and J. Baldridge. A task in a suit and a tie: paraphrase generation with semantic augmentation. *CoRR*, abs/1811.00119, 2018.

[51] S. Wubben, A. van den Bosch, and E. Krahmer. Paraphrase generation as monolingual translation: Data and evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*, 2010.

[52] N. Yaghmazadeh, Y. Wang, I. Dillig, and T. Dillig. Sqlizer: Query synthesis from natural language. *Proc. ACM Program. Lang.*, 1(OOPSLA):1–26, Oct. 2017. doi: 10.1145/3133887

[53] B. Yu and C. T. Silva. Flowsense: A natural language interface for visual data exploration within a dataflow system. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1–11, Jan 2020. doi: 10.1109/TVCG.2019.2934668

[54] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman, Z. Zhang, and D. Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3911–3921. Association for Computational Linguistics, Brussels, Belgium, Oct.-Nov. 2018. doi: 10.18653/v1/D18-1425

[55] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS15, p. 649657. MIT Press, Cambridge, MA, USA, 2015.

[56] V. Zhong, C. Xiong, and R. Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning, 2017.

[57] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau. A c-lstm neural network for text classification, 2015.