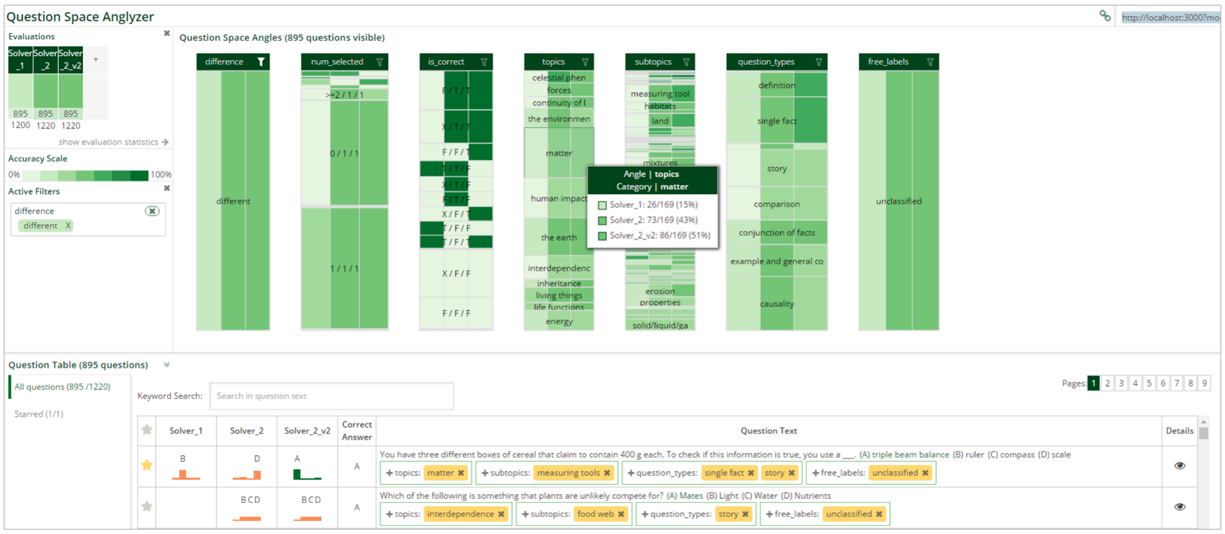
QSAnglyzer: Visual Analytics for Prismatic Analysis of Question Answering System Evaluations

Nan-Chen Chen[*]
University of Washington

Been Kim[†]
Allen Institute for Artificial Intelligence



	Solver_1	Solver_2	Solver_2_v2	Correct Answer	Question Text	Details
★	B	D	A	A	You have three different boxes of cereal that claim to contain 400 g each. To check if this information is true, you use a ___. (A) triple beam balance (B) ruler (C) compass (D) scale +topics: matter ✖ +subtopics: measuring tools ✖ +question_types: single fact ✖ story ✖ +free_labels: unclassified ✖	👁
☆		BCD	BCD	A	Which of the following is something that plants are unlikely compete for? (A) Mates (B) Light (C) Water (D) Nutrients +topics: interdependence ✖ +subtopics: food web ✖ +question_types: story ✖ +free_labels: unclassified ✖	👁

Figure 1: The QSAnglyzer interface consists of three basic panels: 'Evaluations' (top left), 'Question Space Angles' (top right), and 'Question Table' (bottom) panels, following the well-known visualization mantra: "Overview first, zoom and filter, details on demand.

Abstract

Developing sophisticated artificial intelligence (AI) systems requires AI researchers to experiment with different designs and analyze results from evaluations (we refer this task as *evaluation analysis*). In this paper, we tackle the challenges of evaluation analysis in the domain of question-answering (QA) systems. Through in-depth studies with QA researchers, we identify tasks and goals of evaluation analysis and derive a set of design rationales, based on which we propose a novel approach termed *prismatic analysis*. Prismatic analysis examines data through multiple ways of categorization (referred as *angles*). Categories in each angle are measured by *aggregate metrics* to enable diverse comparison scenarios.

To facilitate prismatic analysis of QA evaluations, we design and implement the *Question Space Anglyzer* (QSAnglyzer), a visual analytics (VA) tool. In QSAnglyzer, the high-dimensional space formed by questions is divided into categories based on several angles (e.g., topic and question type). Each category is aggregated by accuracy, the number of questions, and accuracy variance across evaluations. QSAnglyzer visualizes these angles so that QA researchers can examine and compare evaluations from various aspects both individually and collectively. Furthermore, QA researchers filter questions based on any angle by clicking to construct complex queries. We validate QSAnglyzer through controlled experiments and by expert reviews. The results indicate that when using QSAnglyzer, users perform analysis tasks faster ($p < 0.01$) and more accurately ($p < 0.05$), and are quick to gain new insight. We discuss how prismatic analysis and QSAnglyzer scaffold evaluation analysis, and provide directions for future research.

[*]e-mail: nanchen@uw.edu
[†]e-mail: beenkim@csail.mit.edu

Keywords: visual analytics; visualization; interactive visualization; question answering; multi-experiment analysis; visual comparison; visual exploration; prismatic analysis

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—

1 Introduction

In the artificial intelligence (AI) era, complex AI systems such as conversational agents, self-driving cars, and recommendation systems are part of daily life. Various reports indicate that AI plays an increasingly influential role in human society [18, 23]. Major technology companies [1, 12] as well as government agencies [6] are putting increasing emphasis on AI applications.

The development of a sophisticated AI system is a challenging process. AI researchers experiment with different AI system architectures and evaluate their performance on the target tasks. As components in the system interact with each other, it can be difficult to make sense of evaluations and find insight to improve the system. For example, a question answering (QA) system, a type of AI system that answers questions, may involve information retrieval (IR), natural language processing (NLP), knowledge representation, reasoning, and machine learning (ML) components. All components are complex in their own ways, and can interact with each other in complicated ways such that the impact of changes in one component can propagate to overall changes in behavior. AI researchers may modify one component based on insight gained from one set of evaluations, but discover that the overall behavior of the system changes in unexpected ways. For instance, an NLP component can take the output of an IR component as input, and pass this onto an ML component. Small changes in the IR component may thus propagate to the ML component, and lead to unanticipated changes in overall behavior.

Traditional metrics for system performance (e.g., task accuracy) often fail to provide useful and actionable insights for AI researchers. For instance, two versions of a QA system can both achieve 50% accuracy by answering completely different sets of questions correctly. One version may answer all the questions starting with 'what' correctly, whereas the other version may get all the 'why' questions right. More complex patterns of the strengths and weaknesses of each version are challenging to discover and track. The burden for AI researchers is to analyze evaluations to understand how changes impact system behavior, and then find ways to enhance the system.

One way to aid this investigation of evaluations is to introduce categorization. Showing finer-grain patterns may provide deeper insight than showing overall metrics. The challenge is that researchers do not know a priori the best way to categorize evaluations. In addition, the best insight to improve the system may require iterative filtering on various types of categories. The complexity of such exploratory queries can further impose burdens on researchers.

In this paper, we tackle the challenge of evaluation analysis in the domain of QA systems. We work closely with QA researchers to identify their goals and tasks in analyzing QA evaluations, and derive a set of design rationales that support their workflows. These in-depth studies with QA researchers lead to our proposed novel approach termed *prismatic analysis*. This approach examines data from multiple angles, where each angle shows a different way to divide the input data into categories. Categories are then measured by aggregate metrics for further investigation. With this approach, users can perform various comparison scenarios, such as between-category, between-angle, and between-evaluation comparison. In addition, the approach works with data subsets as well, so QA researchers can conduct analysis even after filtering.

Although multiple angles of categories can enable more diverse pattern discovery, it can also be overwhelming to researchers. To support the prismatic analysis of QA evaluations, we design and implement a visual analytics (VA) tool called the *Question Space Anglyzer* (QSAnglyzer). In QSAnglyzer, questions in evaluations constitute a high-dimensional space in which each question is a point. The question space can be divided into categories based on several angles, such as topics and question types. The aggregate metrics for each category that are chosen based on the design rationales include accuracy, the number of questions, and accuracy variance across evaluations. QSAnglyzer visualizes these angles and aggregate metrics of categories using color, height, and order. The QSAnglyzer design enables QA researchers to examine and compare evaluations from various angles individually and collectively. Furthermore, QA researchers filter questions based on any angle by clicking, thus constructing complex queries visually. With controlled experiments and expert reviews, we validate that the tool helps users perform analysis task more quickly ($p < 0.01$) and with greater accuracy ($p < 0.05$), and also generates insight in a short amount of time.

The major contributions of our work are as follows:

- We identify the goals and tasks of QA researchers in analyzing evaluations to derive design rationales. We highlight the need to further study the workflows of complex AI system development.
- We propose prismatic analysis, a novel approach for multi-angle categorization and shared aggregate metrics for category comparison and connection, which enables finer-grain pattern discovery.
- We design and implement QSAnglyzer, a visual analytics system for performing prismatic analysis on QA system evaluations, and validate its effectiveness. We discuss how prismatic analysis and QSAnglyzer scaffold evaluation analysis, and we provide directions for future research to extend the tool and apply prismatic analysis to other AI domains.

2 Background and Related Work

2.1 Question Answering Systems

The QA system is a type of complex AI system that gained recognition when IBM's QA system, *Watson*, appeared on the Jeopardy! game show [7] in 2011. The subject of study now for four decades, QA is one of the oldest AI problems. Numerous QA systems have been proposed to answer short factoid questions [3, 7, 10], standardized tests in geometry [19], and mathematics [11]. Some QA systems are designed to answer multiple-choice questions (e.g., [5, 27]), whereas some are designed for direct-answer questions (e.g., [26, 34]).

Typical QA systems can be roughly divided into two types: IR-based and knowledge-based [8, Chap 28]. IR-based QA systems rely on large sets of unstructured corpora and use IR techniques to search for sentences that contain words found in the questions. In contrast, knowledge-based QA systems are built upon structured knowledge (e.g., lists of tuples for relations between semantically typed entities) for reasoning. The knowledge base is usually smaller and more organized than IR corpora, and the knowledge can be chained to infer unseen facts.

In addition, in complex AI systems, ensemble methods are often used to combine results from a set of weaker AI systems to generate the final outputs. Likewise, a QA system may comprise a set of *solvers* that answer questions, and use ensemble methods to produce the system's final answers.

2.2 VA for Multi-Experiment Result Analysis

Conducting multi-experiment result analysis, including comparison and exploration, is one of the most common usages of visualization in scientific domains. For instance, Nocke et al. propose a series of visualization views for climate science simulation output comparison [17]. Wilson and Potter focus on visualizing ensemble simulation data for weather forecasting and support side-by-side comparisons [30]. Malik et al. utilize visualization to inspect sets of 3D X-ray computed tomography images where device parameters varied when taking the images [13]. Although such applications are common in scientific domains, they do not directly apply to QA system design for three reasons. First, most above-mentioned work focuses on simulation data from the same models with different parameter settings, which is not the case in QA system design. Tuning parameters is only one type of change that QA researchers make to their solvers. Often times the changes are more complex, such as adding more knowledge, changing language processing algorithms, and so on. Furthermore, there is typically more than one type of solver, and they all work very differently from each other; thus there is no intuitive way to visualize the parameter spaces across these heterogeneous solvers. Such unique characteristics of QA systems compose a different design space for visualization from the typical comparative visualization used in scientific domains.

2.3 VA for Inspecting Computational Models

Inspection of computational models, such as ML models, is a growing topic in the field of human-computer interaction (HCI) and visualization (VIS) research. Two key goals of this direction are to better understand complex black-box models, and to reduce the effort required to construct and tune models. The former goal aims to make it easier for model designers to pin down potential issues with the complex computational process, whereas the latter provides clues for improvement of models. For instance, Smilkov and Carter demonstrate an interactive visualization that shows the process of neural networks (NN) [21] so that model designers can see how a NN model progresses over time. Various work in classifier weight tuning [9, 24] or feature ideation [4] also demonstrates how visualization can simplify efforts to construct a better model. Although QA and other complex AI systems also contain computational models, enhancing the performance of such complex systems requires more than focusing on specific models. Therefore, we focus on the workflow level of the development process, and specifically focus on building analytics that support evaluation analysis.

2.4 Multi-view Learning and Multiple clustering Solutions

Prismatic analysis can be related to multi-view learning and multiple clustering solutions. Multi-view learning is considering a machine learning problem from multiple views (e.g., multiple sources or different feature subsets) since the target learning problem cannot be described using a single view [32, 33]. Multiple clustering solutions focus on using multiple ways to cluster data points [14], and have been used in subspace analysis of high-dimensional data space [25]. Like multi-view learning, prismatic analysis also emphasizes the need to consider multiple views (here referred to as 'angles'), but it does not focus on learning problems per se. In addition, multi-view learning does not require defining categories in each view. In contrast, for multiple clustering solutions, the goal is to define categories (i.e., clusters) in many different ways: this is similar to the idea of prismatic analysis. However, prismatic analysis can be positioned as the next step after finding multiple clustering solutions. The results from multiple clustering solutions can be fed into prismatic analysis. Moreover, multiple clustering solutions focus on finding categories, whereas prismatic analysis emphasizes comparisons and making connections between categories. Thus, defining aggregate metrics is also a critical step of prismatic analysis, which is not a part of multiple clustering solutions. In Section 4, we describe in detail prismatic analysis.

3 User Requirement Analysis

In this section, we describe user requirements and design rationales derived through in-depth user studies with QA researchers.

3.1 Identifying User Requirements

To identify user requirements in analyzing QA system evaluations, we worked with a research team with about 20 people who built a QA system at a research institute in the USA for six months. The formative study mostly happened in the first month of the project. During the formative study, we informally interacted with the team in a daily basis, and we conducted a series of interviews with five QA researchers who are primary solver developers. All of the interviewees hold PhD degrees and have been working on the project as research scientists full time for more than a year, and their experience in the related fields ranges from 10 years to 33 years.

The QA system consists of multiple solvers for answering standardized science exam questions. The system uses an ensemble method to combine answers from the solvers, and returns a final answer for each question. For simplicity, we hereafter use 'solvers' to refer to both solvers and the QA system, as the QA system can be considered an ensemble solver that outputs answers based on all solvers' answers.

3.1.1 QA Evaluation Overview

To develop and test the QA system, the QA researchers ran solvers on a set of evaluation questions. When we worked with the QA researchers, their set contained about 1,200 multiple-choice questions, and they sometimes ran evaluations on a subset of the whole question set. They had another question set contained direct-answer questions, but in our study we focused on multiple-choice questions as they were the center of solver development at the time of the study.

To evaluate a solver, the QA researchers run the solver on an arbitrary set of questions. The solver takes each question's description and answer choices, and produces a confidence score for each choice. The choice with the highest confidence score is taken as the solver's final answer to the question. Sometimes, confidence scores of choices are tied, and the solver returns more than one choice as its answer to a question. Currently, QA researchers measure the solver's overall accuracy in answering these questions. In addition to single solver evaluation, QA researchers also run multiple solvers together as an ensemble solver. The output of an ensemble of solvers combines individual solvers answers and returns an answer.

3.1.2 QA Researcher Goals and Tasks

From interviews and informal interactions with the QA researchers, we noted three primary goals for them when analyzing evaluations:

G1. Look for ways to improve a solver. When they analyze evaluations, one key goal is to seek opportunities to improve the solver. Example ways include examine whether a solver has enough data (e.g., knowledge) to answer a question, or whether other dependencies, such as entailment service, impact solver performance.

G2. Examine if new changes to a solver improve performance. After QA researchers modify a solver, they evaluate whether the changes are helpful. As they often have a hypothesis in mind about how the changes may work, they also examine whether the newer version of a solver behaves as expected, and whether any unexpected behavior appears.

G3. Understand the strengths and weaknesses of solvers. Since the final QA system combines the answers of all solvers, it is important to understand the strengths and weaknesses of solvers. This understanding provides opportunities for a researcher to better contribute to overall system performance.

We then collected and compiled the following common tasks by observing how QA researchers investigate evaluations to achieve the above goals. For each task, we note in parentheses its corresponding goals.

T1. Compare two or more versions of a solver for G1 and G2. After modifying a solver, QA researchers usually compare the new version with the original version. They look for questions that the newer version gets right but the older version gets wrong, and vice versa. Sometimes, they investigate questions for which the two versions of the solver have different answers, regardless of the correctness. Since solver development is not a linear process (i.e., versions can branch out), these comparison tasks are sometimes performed on more than two versions at the same time.

T2. Contrast one solver's behavior with that of other solvers (G1, G3). In addition to comparing different versions of a solver, QA researchers also contrast a solver's behavior with that of other solvers to understand solver strengths and weaknesses (G3). QA researchers compare their solvers behavior to that of others as a reference to show increases or decreases in their own solvers performance.

T3. Categorize and/or filter questions (G1, G2, G3). QA researchers often focus on question subsets to improve certain categories (G1), investigate component-wise changes (G2), or identify patterns of solver performance between subsets (G3). To this end, they either try to group the whole set into categories and examine individual categories, or they find a way to retrieve relevant questions (e.g., questions containing certain keywords).

T4. Investigate question descriptions to make sense of a solver's behavior (G1). QA researchers usually read question text and answer choices first to see if they can come up with a hypothesis for the solver's correct or incorrect answer. This task relies on their past experience in developing the solver and debugging similar questions. QA researchers do not always have an idea about why solvers behave in certain ways for a question. Therefore, they have to closely investigate the question to discover situations where, for instance, a solver is being misled by a particular word and thus exhibits extremely different behavior for two very similar questions.

T5. Inspect how a solver answers a question (G1, G2). To know more about how a solver answers a question, QA researchers program solvers to produce intermediate output (e.g., describing the reasoning process) so that they can quickly make sense of what may have happened. They also have solver-specific tools to drill down into detailed steps.

T6. Find insight that can improve a large subset of questions at once (G1). One QA researcher noted, "We want to make changes that can fix (solvers' answers to) many questions, not just one at a time." In order to efficiently gain insight, they typically investigate subsets of questions that have shared properties and that are not too small (i.e., fewer than 10 questions). For instance, they examine questions that belong to the topic 'adaptation' to figure out what knowledge should be added to the knowledge base to answer questions on this topic.

T7. Look for questions for which solvers return no answers, or multiple answers (G1). To improve a solver, one common approach taken by researchers is to investigate questions for which the solver returns no answers, or multiple answers. For these questions they seek ways to modify the solver to break ties between choices, or cause the solver to return a single answer.

T8. Share their findings and insights with other researchers (G1, G3). Researchers sometimes share findings from evaluation analyses with each other. In particular, when discussing solver strengths and weaknesses, spreadsheets and screenshots are popular ways to illustrate their findings.

3.2 Preliminary Investigation: Sunburst Visualization

During our user requirement analysis, the QA researchers expressed interest in utilizing the three types of categories they manually labeled: topic, subtopic, and question type (To be consistent with later sections, we term each categorization an *angle*. Note that angles do not refer to degrees encoded within the Sunburst). Thus, as a way to crystallize user requirements, we created an interactive Sunburst visualization [22] using the three angles (Fig. 2) . In this visualization, a Sunburst corresponds to an evaluation. The three angles were arranged hierarchically. Each layer of the Sunburst consisted of the categories within one angle. The color indicated the solver accuracy in the corresponding categories in an evaluation. The accuracy was bucketed into 10 levels of the red-green scale shown in the legends on the top left. All Sunbursts were linked together such that as users interacted with one of the Sunbursts by clicking a category in any level, the other Sunburst views were updated to reflect the change.

3.2.1 Challenges

We presented the visualization to the QA researchers and conducted contextual inquiries. Although researchers reacted positively and commented that some solver patterns confirmed their understanding of the solver strengths and weaknesses, they indicated a few challenges and issues when using this Sunburst visualization for their analysis:

C1. Difficulty when comparing more than two evaluations. As tasks T1 and T2 indicate, QA researchers need to compare more than two evaluations. We attempted to present more than two Sunbursts to researchers in a single view: Although the visualization did provide an overview of the results of multiple evaluations, the more Sunbursts that were shown (i.e, the more evaluations loaded), the larger area they occupied on the screen. This makes it more difficult to conduct complex analysis and comparisons.

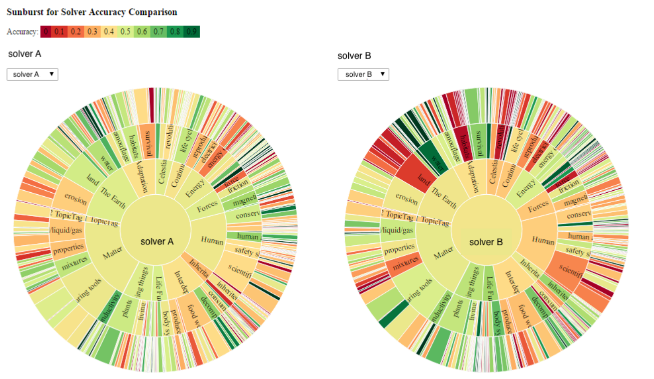


Figure 2: Sunburst visualization utilizing three angles: topic, subtopic, and question type, the three question categories. Each Sunburst layer consists of the categories within an angle. Color indicates solver accuracy for the corresponding categories in an evaluation.

C2. Unidentifiable categories in outer layers. Although we used only a three-level hierarchy in this Sunburst visualization, the researchers still found it difficult to see and identify categories in the outer layers. Although they could filter categories in outer layers of the hierarchy, finding the category they sought to investigate was challenging. This problem is further complicated by additional angles, which add more layers to the Sunbursts.

C3. Complex query requirements. Though the Sunburst visualization provided an overview of the evaluations, QA researchers found it difficult to perform more complex queries. For example, they may want to filter on multiple angles, such as questions that belong to the topic 'Matter' or 'The Earth', but that also belong to the question type 'Definition' or 'Story'. Such complex queries are difficult to perform using the Sunburst visualization.

3.3 Design Rationales

Given the identified goals and common tasks of users, as well as the challenges found in our preliminary investigation of the Sunburst visualization, we present the following design rationales that led to our focus on prismatic analysis and the QSAnglyzer design.

R1. Take questions as the center of analysis. When QA researchers analyze evaluations, questions are the center of analysis. In most of their tasks, they directly investigate questions (T3, T4, and T6) or solver behavior with regard to questions (T5 and T7). In addition, questions are the only aspect of the QA system shared between all solver evaluations. Furthermore, even though solvers can be very different from each other, and not every researcher knows all the solvers, they can still communicate and share insights about the questions (T8).

R2. Prioritize accuracy, the number of questions, and accuracy variance as key variables. When researchers modify a solver, their goal is to increase accuracy (G1). When they investigate a subset of questions, they want to find a large subset to ensure that changes made based on the set have a broad impact (G1 and T3). In addition, when studying solver strengths and weaknesses (G3 and T2), they look for a set of questions for which solvers have highly different accuracy. Therefore, we consider accuracy, the number of questions, and accuracy variance as the high-priority variables for design.

R3. Support categorization. Researchers group questions into categories when focusing on a subset of questions (T3), comparing between versions of a solver (T1), or contrasting solvers (T2). Thus, we observe that categorization can be valuable in revealing patterns in evaluations.

R4. **Consider diverse comparison scenarios and complex queries**. QA researchers need support for diverse comparison scenarios and complex queries. They compare between versions of solvers (T1) and different solvers (T2). They also want to determine whether a solver's performance is worse in one category than in another (T3), to decide which category should be improved (G1). Moreover, they also compare whether two categories have different performance patterns across solvers (T3) to identify strengths and weaknesses (G3). These comparison scenarios can take place after search and filtering (T3 and R5). Therefore, it is necessary to have a design that supports any combination of the above comparison scenarios.

R5. **Enable filtering and search**. As researchers often drill down into a subset of questions (T3), our VA design should include both filtering and search functionalities.

R6. **Show multiple angles at the same time**. QA researchers often want to examine evaluations from multiple angles. For example, they may want to find questions for which two versions of a solver produce different answers (T1). They may also want to see for which topics solver accuracy differs the most (T2). Hence, it is critical to show multiple angles of evaluations. In addition, from challenges C2 and C3 we learn that presenting different angles as layers in a hierarchy may not be ideal, and thus we suggest presenting multiple angles in parallel.

R7. **Ensure scalability**. As there may be more than two sets of evaluations involved in the analysis (C1), it is essential that the design scales up. In our case, we aim to support at least eight sets of evaluations since the QA system contains eight solvers. Additionally, we need to ensure that the design scales to multiple angles (R6).

R8. **Make different levels of evaluation details available**. The design should incorporate various levels of evaluation details, including high-level metrics like accuracy, the description and choices of questions (T4), solver confidence scores on each choice of a question (T7), and intermediate outputs (T5).

R9. **Accommodate existing workflows**. Our VA tool design should fit into the QA researchers' existing workflows so that they can use the tool together with their solver-specific tools (T5), and discuss and share findings with each other (T8).

4 Prismatic Analysis

The study of QA researchers' current evaluation analysis workflows underlies our proposal of *prismatic analysis*. Although this is in the context of QA systems, we argue that the approach can be applied to other types of complex AI system evaluation analysis. Therefore, we start from a generalized formal definition of prismatic analysis using mathematical notation:

Given a set of data points that constitute a high-dimensional *data space* $\mathcal{D}$, we define a set of categorization functions $\mathcal{F}$. Each function $f_i \in \mathcal{F}$ divides $\mathcal{D}$ into a set of categories $\mathcal{C}_i$ from the i-th angle. In addition, we define a set of aggregate metrics $\mathcal{M}$. Each metric $m \in \mathcal{M}$ represents aggregated information of data points in scalar. As a result, we can compare categories within an angle or between angles based on the set of aggregate metrics. Furthermore, some of the metrics may have multiple measurements, and thus we can also compare these measurements within a category. Last, we can filter to a subset of $\mathcal{D}$ and conduct the same analysis.

We call this approach 'prismatic analysis' because when we consider the data space $\mathcal{D}$, the categorization functions $\mathcal{F}$ act like a set of prisms. These prisms are oriented at different angles toward the space so that we see the space divided in various ways. This results in multiple ways to group the data points into categories. Then the set of aggregate metrics $\mathcal{M}$ summarizes all the categories in the same way regardless of angles; this aids us in making comparisons within and between categories in an angle, as well as making connections between angles.

We then describe how prismatic analysis can be applied to analyze QA evaluations: As questions are the center of QA evaluation analysis (R1), the *question space*, the high-dimensional space formed by questions, corresponds to the data space $\mathcal{D}$. The question space can be divided into categories from multiple angles (each angle corresponding to a categorization function in $\mathcal{F}$), such as by topics or by question types. This implements our design rationales R3 and R6. The aggregate metrics $\mathcal{M}$ here include accuracy (denoted as M_{acc}), the number of questions (M_{num}), and accuracy variance (M_{var}) between evaluations, as suggested by design rationale R2. The M_{acc} metric can have multiple measurements when we load multiple evaluations. Moreover, prismatic analysis allows us to filter subsets of questions to conduct the same analysis, which fulfills design rationale R5.

Prismatic analysis is not limited to QA system evaluations. This approach can be applied to other types of complex AI system evaluation analysis by identifying the data space $\mathcal{D}$, categorization functions $\mathcal{F}$, and aggregate metrics $\mathcal{M}$. We further discuss this point in Section 7.4.

Although prismatic analysis captures many analytical tasks performed by QA researchers, it can be overwhelming to have multiple angles and metrics in pure text (e.g., labels and numbers). In addition, design rationales R7, R8, and R9 are not directly supported by prismatic analysis. This leads to the design of QSAnglyzer, our visual analytics tool introduced in the next section.

5 QSAnglyzer

In this section, we describe the design of QSAnglyzer, which is implemented as a JavaScript web application using React.js and D3.js. All data are stored in a PostgreSQL database and can be accessed through web service calls (implemented in Node.js). QSAnglyzer aims to support prismatic analysis of QA evaluations; its design follows the rationales we derived in Section 3.3. To be more specific, QSAnglyzer enables QA researchers to load evaluations into the tool, and supports visual comparison and interactions based on the framework of prismatic analysis. The system treats each evaluation as a dataset in which each question is a data point. Results from different evaluations are different dimensions of the data points. The system provides seven default angles ($\mathcal{F}$) to divide questions into groups; we explain the details of each angle in Section 5.1. After the user loads an evaluation on the system or performs filtering or search, the system automatically calculates the aggregate metrics (M_{acc}, M_{num}, and M_{var}) for each category. The metric M_{acc} has N measures, one for each evaluation, where N is the number of evaluations loaded on the system.

5.1 Angles

By default, the system provides seven angles that can be divided into three types: question-related angles, evaluation-related angles, and free labels. These angles are chosen and defined based on what we found QA researchers need, though we envision that more angles will be defined in the future. Question-related angles utilize the three sets of manually labeled categories from the QA researchers (topics, subtopics, and question types) used in Section 3.2.

Evaluation-related angles are a set of angles automatically derived from evaluations. These include correctness (is_correct), agreement of answers (difference), and the number of selected choices (num_selected). The 'is_correct' angle categorizes questions based on whether a selected choice was correct (T) or not (F), or having no answer (X). For multiple evaluations, the categories are correctness permutations across evaluations. For example, when the system is loaded with three evaluations, the ‘F / T / T’ category refers to questions for which the first evaluation is incorrect but the second and third evaluations are both correct.

The ‘difference’ angle divides questions into two categories: ‘same’ and ’different’. Questions for which all evaluations have the same answer belong to the former; the rest belong to the latter. A question may have the same answers in two evaluations, but as long as there is another evaluation that has a different answer, it belongs to the ‘different’ category.

The ‘num_selected’ angle aims to support task T7 by categorizing questions based on the number of selected choices. For simplicity, we treat questions with more than two choices as being in the same group; thus each question can only have ‘0’, ‘1’, or ‘>=2’ selected choices. Similar to the ‘correctness’ angle, in the case of multiple evaluations, the categories are permutations across evaluations. For instance, when there are three evaluations, the category ‘>=2 / 1 / 1’ includes questions for which the first evaluation selected more than one choice, and the other two selected only one choice.

The ‘free_labels’ angle is designed for users to assign customized categories to questions. Users label the questions in the bottom ‘Question Table’ panel. We describe this process in Section 5.2.3.

5.2 Interface Design

As shown in Fig. 1, the QSAnglyzer interface consists of three basic panels: the top left ‘Evaluations’, the top right ‘Question Space Angles’, and bottom ‘Question Table’ panels. This design follows the visualization mantra: “Overview first, zoom and filter, details on demand [20].” The ‘Evaluations’ panel provides the most high-level overview of the evaluations, and helps users keep track of how many questions have been filtered by showing the partial bars. In the ‘Question Space Angles’ panel, users perform prismatic analysis; this panel allows users to filter and zoom into a finer set of questions. Finally, the ’Questions Table’ panel shows details related to questions and solver behavior, and supports keyword search and question bookmarking (i.e., starring). This design fulfills our design rationale R8. In addition to the basic panels, users obtain a link to share the current view of the tool by clicking the link icon on the top right corner of the tool, which follows design rational R9. Below, we explain each component of the QSAnglyzer and the design decisions for visual encoding and tool interaction.

5.2.1 Top Left Panel: A Visual Summary of Evaluations

In the top left ‘Evaluations’ panel, users add or remove evaluations to visualize by providing URLs (the URLs link to their existing database APIs). Users assign a name to the evaluation, or the system automatically names the evaluation based on the solver. We represent each set as a colored vertical bar, where the color indicates the overall accuracy (dark green represents high accuracy). The maximum height of the bars corresponds to the highest number of questions loaded to the tool, and the height of the individual bars are proportional to the number of questions in each evaluation (labeled in text under the bars). When hovering over the bar, a tooltip pops up to show the exact accuracy.

These bars provide a visual summary for evaluations, and are updated when the user interacts with the tool. When a user filters a set of questions, a partial bar appears reflecting this filtering action. The color of the partial bar represents the accuracy for the filtered set of questions. If the number of questions varies across evaluations, the system automatically filters the subset of questions that all evaluations contain.

Design choices Although color is not the best choice for encoding quantitative values such as accuracy, it is a visual attribute supported by human pre-attentive processing [28]. In addition, since length is a recommended visual attribute for showing quantitative data [16], we use it to encode the number of questions. The two design choices are also used in the ‘Question Space Angles’ panel.

5.2.2 Top Right Panel: Question Space Angles

The *Question Space Angles* panel on the top right supports prismatic analysis as a metaphorical representation of the question space. Each box on the panel represents an angle (an $f_i \in \mathcal{F}$) and its categories ($\mathcal{C}_i$), and each category is represented by a horizontal slice. The height of a slice corresponds to the number of questions (M_{num}) in this category, where the height of an entire angle box corresponds to the number of visible questions (i.e., all questions or remaining questions after filtering). For simplicity, hereafter we directly use ‘angles’ to refer to these boxes, and ‘categories’ to refer to the horizontal slices.

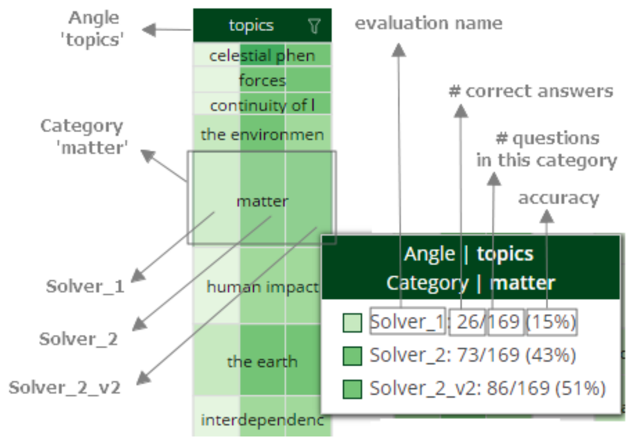
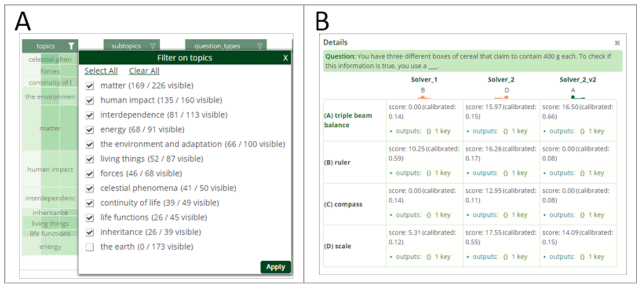


Figure 3: The design of an angle. The header shows the name of the angle. Each slice is a category within the angle, and its height corresponds to the number of questions in the category. The vertical segments represent evaluations, and the colors indicate how accurate the solver in the evaluation is on the questions in the category. When hovering over a category, a tooltip appears, providing detailed numbers.

Fig. 3 illustrates the parts of an angle. When there are N evaluations, every category is divided into N vertical segments. Each segment represents one evaluation. For example, in Fig. 1, as the tool is loaded with three evaluations, each category has three segments. The order of the segments follows the order in the ‘Evaluations’ panel. The color of each segment reflects the accuracy (M_{acc}) of the solver in an evaluation on the questions within the category.

When the user clicks a category within an angle, a filtering action is triggered, and all angles are updated based on the remaining questions. When users hover over a category, a tooltip appears (Fig. 3), providing detailed information such as the number of questions in the category, the number of correctly-answered questions in each evaluation, and the corresponding accuracy. When users seek to filter more than one category, they use the filter box shown in Fig. 4A. If a filter is applied to an angle, the filter icon on its header turns white. The ‘Active Filters’ panel on the left lists the currently applied filters.

Categories within an angle are sorted with respect to variances in accuracy (M_{var}) across evaluations (i.e., the first category on the top has the highest variance across evaluations). In the filter box, however, we sort the categories with respect to the number of questions (M_{num}). As we have observed cases in which both the order of M_{num} and M_{var} can be useful, we use different ordering mechanisms as a quick way to provide the two types of information without re-ordering.


A

topics | subtopics | question_types

Filter on topics

Select All | Clear All

- [x] matter (169 / 226 visible)
- [x] human impact (135 / 160 visible)
- [x] interdependence (81 / 113 visible)
- [x] energy (68 / 91 visible)
- [x] the environment and adaptation (66 / 100 visible)
- [x] living things (52 / 87 visible)
- [x] forces (46 / 68 visible)
- [x] celestial phenomena (41 / 50 visible)
- [x] continuity of life (39 / 49 visible)
- [x] life functions (26 / 45 visible)
- [x] inheritance (26 / 39 visible)
- [ ] the earth (0 / 173 visible)

Apply

B

Details

Question: You have three different boxes of cereal that claim to contain 400 g each. To check if this information is true, you use a ___.

	Solver_1	Solver_2	Solver_2_v2
	B	D	A
(A) triple beam balance	score: 0.00 (calibrated: 0.14) ▸ outputs: {} 1 key	score: 15.97 (calibrated: 0.15) ▸ outputs: {} 1 key	score: 16.50 (calibrated: 0.66) ▸ outputs: {} 1 key
(B) ruler	score: 10.25 (calibrated: 0.59) ▸ outputs: {} 1 key	score: 16.26 (calibrated: 0.17) ▸ outputs: {} 1 key	score: 0.00 (calibrated: 0.08) ▸ outputs: {} 1 key
(C) compass	score: 0.00 (calibrated: 0.14) ▸ outputs: {} 1 key	score: 12.95 (calibrated: 0.11) ▸ outputs: {} 1 key	score: 0.00 (calibrated: 0.08) ▸ outputs: {} 1 key
(D) scale	score: 5.31 (calibrated: 0.12) ▸ outputs: {} 1 key	score: 17.55 (calibrated: 0.55) ▸ outputs: {} 1 key	score: 14.09 (calibrated: 0.15) ▸ outputs: {} 1 key

Figure 4: (A) The filter box of the 'topics' angle. The user can filter on more than one category using the check boxes on the left. The categories are sorted with respect to the number of questions. (B) The panel shows the details of the solvers' answers to a question. By unfolding the 'outputs', users examine a solver's intermediate outputs for an choice.

Design choices We chose to use a box to represent an angle and show all angles in parallel to reflect challenge C2 and design rationales R6 and R7. Such a design enables the easy adding and removing of angles, as well as changing their order. In addition, putting evaluations as segments next to each other greatly facilitates comparisons between evaluations and reflects challenge C1. We easily scale up to nine evaluations (R7). In addition, we chose to use orders to encode M_{var}. Although M_{var} is a quantitative variable, the exact values and the difference between values are not critical for analysis. Therefore, we assign a ordinal encoding to the variable. We also use orders to show M_{num}. This is an additional encoding to the height encoding for this variable, which is suitable for ordinal encoding.

5.2.3 Bottom Panel: Question Table

The 'Question Table' panel at the bottom provides finer-grain details of solver evaluations. By default, this panel is collapsed. Every row in the table corresponds to a question and solvers' answers in the evaluations. The first column enables users to 'bookmark' (or 'star') questions, which can then be viewed as the bookmarked questions in the 'Starred' tab. The second and following columns of the table show the solvers' choices and confidence scores in each evaluation. The order of the evaluations is consistent with both the order in the 'Evaluations' panel as well as that in the categories.

Bar charts in these columns represent the normalized confidence scores (between 0 and 1) returned from each evaluation, where the x-axis is the choices and the y-axis is the scores. When users mouse over the bars, a small tooltip appears with the normalized and raw scores. The letter above each bar chart represents the choice with the highest confidence score (that is, the selected choice). When a solver returns more than one choice, the letters of all the selected choices are shown above the bar chart, whereas when a solver returns no answer, the grid is empty (like the first column in the second row in Fig. 1). In contrast to the accuracy colors in the top two panels, the colors in these bar charts are binary: green for correctly answered questions, and orange for incorrectly answered questions.

In addition to solvers' answers, each row of the question table also shows the question's membership to each angle's categories (the yellow labels in the bottom panel of Fig. 1). Users change the question's membership of a category by adding or removing labels. This is also where users create and assign customized labels to the 'free_labels' angle.

Currently, each page of the question table contains 100 questions. Users can also perform keyword searches. As with filtering, completed search actions update the interface to reflect the questions returned from the search. If users want to retrieve solvers' intermediate outputs for a question, they click the eye icon in the last column. A pop-up panel (Fig. 4B) appears so that users can compare intermediate outputs (folded under 'outputs' by default).

Design choices The design of this panel is based on the spreadsheets the QA researchers created before the study. Therefore, even though both colors and heights encode different variables from other panels, for the QA researchers these encodings are acceptable. Note that we designed this panel for multiple-choice questions, but the design can be tweaked to show direct-answer questions without the need to change the other panels.

Table 1: Lab study design. We took the number of evaluations ($IV_{NumOfEval}$) and the tool used (IV_{tool}) as the independent variables.

		Tools (IV_{tool})	
		Excel (control)	**QSAnglyzer (treatment)**
# Evaluations ($IV_{NumOfEval}$)	**2**	2-Excel	2-QSAnglyzer
	4	4-Excel	4-QSAnglyzer
	6	6-Excel	6-QSAnglyzer

6 Evaluation

To verify the effectiveness and efficiency of QSAnglyzer, we conducted a lab study with non-experts and expert reviews. The goal of the lab study with non-experts was to evaluate the effectiveness of the visual and interaction design of QSAnglyzer. We defined the tasks in the lab study based on tasks drawn from the requirement analysis. With the expert reviews, we sought to examine whether QSAnglyzer facilitates insight discovery. The two evaluation approaches are designed based on Munzner's nested model [15]. In this section, we describe both studies in detail.

6.1 Lab Study with Non-experts

Since QA researchers commonly utilize spreadsheets in their workflows, we designed a set of Excel spreadsheets that support the same functionalities for prismatic analysis. These functionalities include the use of multiple angles to categorize questions; the use of accuracy, the number of questions, and accuracy variance as aggregate metrics; and filtering. In addition to the interface, we tested whether the number of evaluations loaded in the tools impacts user performance.

6.1.1 Lab Study Design

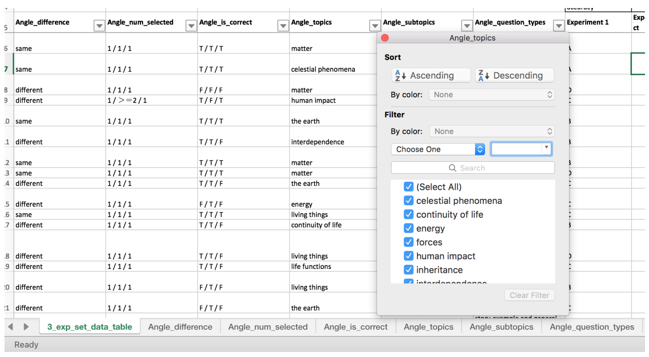
Participants were asked to perform sets of predefined tasks with both QSAnglyzer and the Excel spreadsheets. In addition, we tested both Excel spreadsheets and our tool with 2-, 4-, and 6-solver evaluations (details in Table 1). We took as the independent variables the number of evaluations ($IV_{NumOfEval}$) and the tool used (IV_{tool}). For each condition, the participants performed three tasks; we measured the time spent (DV_{time}) and the accuracy (DV_{acc}) of these tasks. We adapted a within-group study design in which all participants went through all conditions. We used balanced Latin square design [29] to minimize the learning effect.

6.1.2 Data and Tasks

We prepared three sets of evaluations from different solvers or versions of solvers to design tasks. The tasks were selected from the common workflow of QA researchers. Here is an example task and what it was meant to achieve for QA researchers:

"For questions for which the three solvers yield different answers, filter to the category "F/T/F" of the "is_correct" angle. What is the category of "question_type" that has the largest number of questions?" (This task is a combination of tasks T1, T3, and T6.)

The goal of this task is a combination of goals G2 and G3. QA researchers are interested in questions that solvers disagree on, and



	Angle_difference	Angle_num_selected	Angle_is_correct	Angle_topics	Angle_subtopics	Angle_question_types	Experiment 1
6	same	1 / 1 / 1	T / T / T	matter			
7	same	1 / 1 / 1	T / T / T	celestial phenomena			
8	different	1 / 1 / 1	F / F / F	matter			
9	different	1 / >=2 / 1	T / F / T	human impact			
0	same	1 / 1 / 1	T / T / T	the earth			
1	different	1 / 1 / 1	T / T / F	interdependence			
2	same	1 / 1 / 1	T / T / T	matter			
3	same	1 / 1 / 1	T / T / T	matter			
4	different	1 / 1 / 1	T / T / F	the earth			
5	different	1 / 1 / 1	F / T / F	energy			
6	same	1 / 1 / 1	T / T / T	living things			
7	different	1 / 1 / 1	T / T / F	continuity of life			
8	different	1 / 1 / 1	T / T / F	living things			
9	different	1 / 1 / 1	T / T / F	life functions			
0	different	1 / 1 / 1	F / T / F	living things			
1	different	1 / 1 / 1	F / T / F	the earth			

Angle_topics

Sort

Ascending Descending

By color: None

Filter

By color: None

Choose One

Search

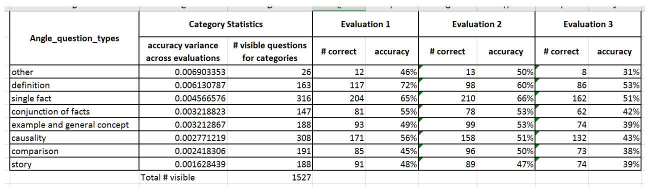
✓ (Select All)
✓ celestial phenomena
✓ continuity of life
✓ energy
✓ forces
✓ human impact
✓ inheritance

Clear Filter

3_exp_set_data_table | Angle_difference | Angle_num_selected | Angle_is_correct | Angle_topics | Angle_subtopics | Angle_question_types

Ready

Figure 5: Screenshot of a highly customized Excel spreadsheet for the baseline condition. Users can perform rich interactions: for example, users can filter to a category or categories, and the summarized performance statistics of each category are pre-calculated and presented.



Angle_question_types	Category Statistics		Evaluation 1		Evaluation 2		Evaluation 3	
	accuracy variance across evaluations	# visible questions for categories	# correct	accuracy	# correct	accuracy	# correct	accuracy
other	0.006903353	26	12	46%	13	50%	8	31%
definition	0.006130787	163	117	72%	98	60%	86	53%
single fact	0.004566576	316	204	65%	210	66%	162	51%
conjunction of facts	0.003218823	147	81	55%	78	53%	62	42%
example and general concept	0.003212867	188	93	49%	99	53%	74	39%
causality	0.002771219	308	171	56%	158	51%	132	43%
comparison	0.002418306	191	85	45%	96	50%	73	38%
story	0.001628439	188	91	48%	89	47%	74	39%
	Total # visible	1527						

Figure 6: Aggregate metrics in the customized Excel spreadsheet for the categories of the 'topics' angle. Upon filtering, aggregate metrics are automatically updated.

specifically look at those question types for which a solver outperforms others.

6.1.3 Excel Spreadsheets

We created highly customized Excel spreadsheets that support prismatic analysis (Fig. 5). For example, as shown in Fig. 6, aggregate metrics of each categories are also pre-calculated and presented. A data table on the first sheet enables users to filter on angles. Upon filtering, aggregate metrics are automatically updated. However, unlike QSAnglyzer, the order of the categories in the spreadsheet is not re-ordered after filtering. We believe this is a reasonable difference and limitation, since automatic re-ordering requires the writing of Excel-specific scripts, which for most users is not an option.

6.1.4 Participants

From the research institute and from the authors' social networks, we recruited 13 non-experts who were not QA researchers but had experience in AI-related fields. The ages of the participants ranged from 21 to 40 (9 male, 3 female, and 1 other gender). Ten held post-graduate degrees in computer science or related fields.

6.1.5 Study Setup and Process

Each participant came to the study room individually. We described the context of the project to them, and then explained the two tools: QSAnglyzer and the customized Excel spreadsheets. Prior to audio and screen recording, the moderator secured participant permission. All tasks were performed on an 28-inch external monitor with a USB mouse and keyboard.

Prior to the formal sessions, the participants were given four practice tasks: two on each tool. We guided the participants in the use of both the Excel spreadsheets and QSAnglyzer, and ensured they

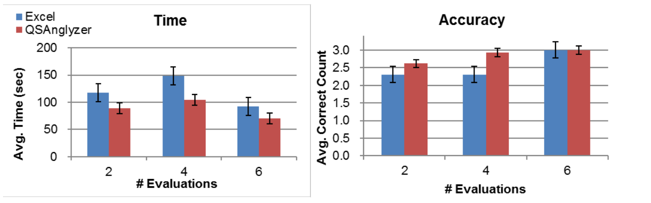


Figure 7: Average time and accuracy for test conditions. Statistical testing showed significant differences between tools (IV_{tool}) on both DV_{time} ($p < 0.01$) and DV_{acc} ($p < 0.05$).

understood the tasks. The formal study consisted of six sessions corresponding to each study condition. In each session, the participants were given one of the two tools preloaded with N evaluations. The number N depended on which condition the session was. During each session, the participants performed three tasks which were randomly selected from a pool of predefined tasks without replacement. Each task had a three-minute time limit. We ensured that participants spent no more than the allotted time on the tasks.

During the practice session, the participants spoke aloud their task answers directly. Once the formal sessions started, the participants were required to submit their answers via a Google form in a window next to the tool. After performing each task, we helped participants restore the tool to its initial state with no filtering and sorting.

6.1.6 Analysis and Results

We analyzed the results of the lab study by a mixed-effects model analysis of variance. The model took the within-group variables (IV_{tool} and $IV_{NumOfEval}$) as factorial effects, but considered each participant as a random level drawn from a population [31]. For each condition, the time (DV_{time}) of each participant was the average time for the completion of three tasks, whereas the accuracy (DV_{acc}) was the number of correctly answered tasks (up to three). The average time and accuracy for each condition are shown in Fig. 7.

Statistical testing showed significant differences between the tools (IV_{tool}) for both DV_{time} ($p < 0.01$) and DV_{acc} ($p < 0.05$), but the difference between the number of loaded evaluations ($IV_{NumOfEval}$) was significant with respect to neither time nor accuracy. There was also no interaction effect between IV_{tool} and $IV_{NumOfEval}$. These results showed that QSAnglyzer more effectively and efficiently supports users performing the tasks. Interestingly, the results did not show significant differences between three levels of $IV_{NumOfEval}$. This may be because the tasks did not emphasize the comparison of multiple evaluations. Even though the results were not significant, during the study we did observe that when more evaluations were given, participants expressed difficulty in finding the target categories.

6.2 Use Cases from Expert Reviews

We invited five QA researchers in the research institute to test and review QSAnglyzer. Each spent an hour testing the tool, and we asked them to think aloud throughout the testing sessions. We preloaded the evaluations of the solvers they developed, and let them explore freely. The researchers all reacted positively and found numerous insights within an hour. They commented that being able to thoroughly examine questions from various angles enabled them to quickly form hypotheses and verify them. They also starred many questions that they were interested in for further inspection. Below we highlight some of use cases and insights they found during the sessions.

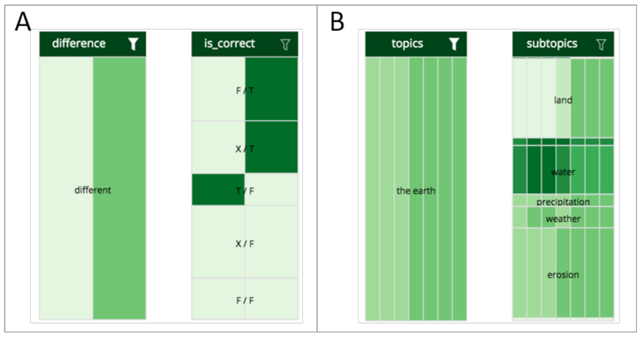


Figure 8: User insights: (A) One of the QA researchers confirmed their solver was improving in the newer version (the right segments) by examining the ‘difference’ and ‘is_correct’ angles. After filtering to the ‘different’ category, the QA researcher was able to draw this conclusion by comparing the number of questions in the ‘X / T’ and ‘F / T’ categories with the number of questions in the ‘T / F’. (B) Another QA researcher discovered an unusual trend over time in the ‘water’ category under the ‘subtopic’ angle. The tool was loaded with seven different versions of a solver (in chronological order from left to right). The researcher filtered to the ‘the earth’ category in the ‘topic’ angle, and found this trend in the ‘subtopic’ angle. After further examination of the intermediate outputs, the QA researcher concluded that the earlier versions had higher accuracy in this category only by chance.

6.2.1 Investigating Gain and Loss of Accuracy after Changes

As noted in goal G2, QA researchers seek to examine how changes to a solver impact its behavior. The researchers performed this task using QSAnglyzer. For example, one researcher first filtered to the ‘different’ category in the ‘difference’ angle, and then investigated the ‘is_correct’ angle (Fig. 8A) by comparing the height of the categories. The researcher then explained: “This (the “is_correct” angle) shows that we have gained more than we lost after adding the new knowledge base... which is a good thing. It also shows that many questions our solver could not answer before (marked as “X” in the categories’ names) now have answers.”

6.2.2 Uncovering Incidentally Correct Questions

In one session, the tool was loaded with the evaluations of seven versions of a specific solver, from left to right in chronological order. Fig. 8B shows a pattern discovered by a researcher in this session; the researcher learned that the solver’s overall performance gradually improved for questions on the topic “the earth” due to new knowledge added to the solver’s knowledge base. However, the researcher noticed that the subtopic “water” did not show the same improvement. By filtering for questions with the “water” category, the QA researcher concluded that the first couple of versions of the solver were answering this set of questions correctly only by chance. This insight was novel and thought-provoking to the QA researcher.

6.2.3 Discovering Previously Unknown Solver Behaviors

During the evaluation sessions, three QA researchers observed solver behavior that surprised them. For instance, because of the “num_selected” angle a researcher realized that an earlier version of his solver chose more than one answer choice for many questions (more than 30). He noted that this pattern is significant since the tie between answer choices had been broken by newer versions of the solver. Another researcher commented that he did not know there were questions where his solver did not choose any answer choice at all. He also noted that he discovered a potential bug using QSAnglyzer: he believed that the sum of all confidence scores of all answers should equal one, which was not always the case.

6.2.4 Relating to Different Solvers

In addition to loading different versions of the same solver, we also showed evaluations of different solvers to the QA researchers. One found that a legacy solver that used an IR-based approach tended to perform better on “story”-type questions (a category of the “question types” angle). Another QA researcher compared two versions of a solver along with a legacy IR-based solver as a reference. By doing so, the QA researcher learned that the solver behaved much like the legacy IR-based solver: “I am interested in questions that used to be wrong, and now are correct, and want to see if the IR-based solver also answered correctly on those questions. (after the exploration and comparison)... I guess my solver now becomes more like an IR-based solver.”

6.2.5 Finding Tricky or Mislabeled Questions

Since QSAnglyzer can be used to quickly form complex queries on different angles, all researchers found questions that they considered tricky or bad questions (e.g., they thought there was more than one correct answer). One found a few mislabeled questions, and was able to use the label modification functions in the ‘Question Table’ panel after the tool was deployed to their work environment.

7 Discussion and Future Work

In this section, we discuss how prismatic analysis and QSAnglyzer scaffold evaluation analysis. Then we discuss potential directions to extend the tool and other future directions of research.

7.1 Scaffolding Evaluation Analysis Workflows

Since QA or other AI systems are often highly complex, generating insights from evaluations requires scaffolding diverse scenarios for comparison and examination. Prismatic analysis provides a foundation that supports such scenarios by multi-angle categorization. The shared aggregate metrics across angles enables diverse scenarios of comparing categories. The QSAnglyzer design further supports the analysis workflows by leveraging the power of visual comparison and interactive filtering.

From the lab study we found that when performing the comparison tasks using QSAnglyzer, users are faster and more accurate. From expert reviews we witnessed QA researchers gain insight into their solvers within one-hour sessions. In addition, when having multiple angles in separate spreadsheets as in our lab study, we found users often lost track and forgot what filters had been applied. In contrast, QSAnglyzer helped QA researchers to maintain context by presenting angles and filters at the same view. This aligns with our design rationale R6.

7.2 Adding Evaluation-based or Automatic Angles

The current QSAnglyzer design includes only seven angles. Though these angles are defined based on the QA researchers’ need, there is space to add more angles. Three directions for future work include: evaluation-dependent, semi-automatic, and fully-automatic angles. Evaluation-dependent angles refer to angles that are specific to an evaluation, unlike the current evaluation-related angles, which treat all evaluations equally. For example, solvers usually have features for intermediate processing: we can group questions based on these features. However, features are evaluation-dependent and may be extremely different across evaluations, which requires further studies for categorization and interface design. Semi-automatic angles are an augmented version of the ‘free_labels’ angle. Although QSAnglyzer supports manually-created angles, researchers commented that they want the system to recommend similar questions when they assign a category label to a question: fully-automatic angles should create categories using clustering algorithms. These three directions are not mutually exclusive. Thus, we can also create evaluation-dependent angles using clustering algorithms, or we can run topic modeling algorithms (e.g., [2]) to create automatically-generated topics. Preliminary investigation in this direction suggests a challenge: questions that are interesting to researchers may drift from time to time. It may be difficult to collect a set of questions to define other categories that researchers seek to investigate in every evaluation. In addition, automatic algorithms may often result in uninterpretable categories, and thus may not be useful to users. We leave addressing these issues for future work.

7.3 Supporting Complex AI System Development

Evaluation analysis is critical in developing complex AI systems, yet it is only part of the development workflow. Many other parts of the workflows are under-studied and could be better supported. For instance, since each solver has many components, we could also build visual analytics to help researchers to understand the interaction between QA system components. In addition, one QA researcher told us that QSAnglyzer is useful when they have developed a solver that is accurate enough. When a solver is in its initial development stage, they may want to focus on getting one or two questions correct, rather than on reviewing overall patterns. These examples indicate the need for more holistic studies on complex AI system development, which could lead to an open design space of visual analytics tools.

7.4 Applying Prismatic Analysis to Other AI Domains

One key reason for using prismatic analysis is that the analysis target (e.g., evaluations) and the goal (e.g., improving a QA system) are complex. This necessitates examination from many angles and categorization to help discover patterns. Therefore, prismatic analysis is suitable not only for analyzing QA system evaluations, but also any complex analysis target that can benefit from multi-angle categorization. For example, a potential use case for prismatic analysis is analyzing evaluations of a recommendation system. We can categorize user profiles from multiple angles, and determine how well a recommendation system performs on various profile categories. However, since in AI systems new development practices keep emerging, more in-depth studies are needed on user workflows to apply prismatic analysis in different contexts. In addition, in the QSAnglyzer design, we focus on three aggregate metrics, one of which (M_{acc}) has multiple measurements. Other contexts may have different aggregate metrics, and thus the design of QSAnglyzer may be adjusted. Using prismatic analysis in different contexts and building visual analytics for different contexts are directions that call for further exploration.

8 Conclusion

Building AI systems is a fascinating but challenging problem to AI researchers, yet most work in HCI and VIS focuses only on the development and investigation of individual models. In this paper, we address the development of complex AI systems in the domain of QA systems. Specifically, we focus on the workflows of multiple-evaluation analysis. We work closely with QA researchers to extract their goals and tasks in evaluation analysis, identifying challenges and design rationales that lead to our proposal of prismatic analysis and QSAnglyzer. Although here this is applied to the QA domain, we envision that prismatic analysis can be applied to other AI domains and so on as the design of QSAnglyzer. This work opens a door and highlights the need for better support for AI development. More studies should further examine the design space and seek opportunities for visual analytics research.

Acknowledgments

We wish to thank all the participants for their time and the anonymous reviewers for their helpful comments. This work is supported by Allen Institute for Artificial Intelligence (AI2).

References

[1] L. Bell. Artificial intelligence is now intel's major focus, 2016. Retrieved from `http://www.alphr.com/the-future/1004845/artificial-intelligence-is-now-intels-major-focus`.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[3] E. Brill, S. Dumais, and M. Banko. An analysis of the askmsr question-answering system. pp. 257–264. Association for Computational Linguistics, 2002.

[4] M. Brooks, S. Amershi, B. Lee, S. M. Drucker, A. Kapoor, and P. Simard. FeatureInsight: Visual support for error-driven feature ideation in text classification. In *Proceedings of the 2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 105–112, 2015. doi: 10.1109/VAST.2015.7347637

[5] P. Clark, O. Etzioni, T. Khot, A. Sabharwal, O. Tafjord, P. Turney, and D. Khashabi. Combining retrieval, statistics, and inference to answer elementary science questions. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016.

[6] Defense Advanced Research Projects Agency. Explainable Artificial Intelligence (XAI), 2016. Retrieved from `http://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf`.

[7] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010.

[8] D. Jurafsky and J. H. Martin. *Speech and language processing*, vol. 3. Pearson, 2014.

[9] A. Kapoor, B. Lee, D. Tan, and E. Horvitz. Performance and preferences: Interactive refinement of machine learning procedures. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012. 00012.

[10] J. Ko, E. Nyberg, and L. Si. A probabilistic graphical model for joint answer ranking in question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 343–350. ACM, 2007.

[11] N. Kushman, Y. Artzi, L. Zettlemoyer, and R. Barzilay. Learning to automatically solve algebra word problems. Association for Computational Linguistics, 2014.

[12] G. Lewis-Kraus. The Great A.I. Awakening, 2016. Retrieved from `https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html`.

[13] M. M. Malik, C. Heinzl, and M. E. Groeller. Comparative visualization for parameter studies of dataset series. 16(5):829–840, 2010. doi: 10.1109/TVCG.2010.20

[14] E. Muller, S. Gunnemann, I. Farber, and T. Seidl. Discovering multiple clustering solutions: Grouping objects in different views of the data. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pp. 1207–1210. IEEE, 2012.

[15] T. Munzner. A nested model for visualization design and validation. 15(6):921–928, 2009. doi: 10.1109/TVCG.2009.111

[16] T. Munzner. *Visualization Analysis and Design*. AK Peters Visualization Series. CRC Press, 2014.

[17] T. Nocke, M. Flechsig, and U. Bohm. Visual exploration and evaluation of climate-related simulation data. In *2007 Winter Simulation Conference*, pp. 703–711, 2007. doi: 10.1109/WSC.2007.4419664

[18] S. Russell, D. Dewey, and M. Tegmark. Research priorities for robust and beneficial artificial intelligence. *AI Magazine*, 36(4):105–114, 2015.

[19] M. J. Seo, H. Hajishirzi, A. Farhadi, and O. Etzioni. Diagram understanding in geometry questions. In *AAAI*, pp. 2831–2838, 2014.

[20] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, pp. 336–343. IEEE, 1996.

[21] D. Smilkov and S. Carter. Tensorflow neural network playground, 2016. Retrieved from `http://playground.tensorflow.org`.

[22] J. Stasko and E. Zhang. Focus+ context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of the 2000 IEEE Symposium on Information Visualization*, pp. 57–65. IEEE, 2000.

[23] P. Stone, R. Brooks, E. Brynjolfsson, R. Calo, O. Etzioni, G. Hager, J. Hirschberg, S. Kalyanakrishnan, E. Kamar, S. Kraus, K. Leyton-Brown, D. Parkes, W. Press, A. Saxenian, J. Shah, M. Tambe, and A. Teller. "Artificial Intelligence and Life in 2030" - One Hundred Year Study on Artificial Intelligence: Report of the 2015-2016 Study Panel. Technical report, Stanford University, Stanford, CA, September 2016. Retrieved from `http://ai100.stanford.edu/2016-report`.

[24] J. Talbot, B. Lee, A. Kapoor, and D. S. Tan. EnsembleMatrix: Interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pp. 1283–1292. ACM, 2009. 00091. doi: 10.1145/1518701.1518895

[25] A. Tatu, F. Maaß, I. Färber, E. Bertini, T. Schreck, T. Seidl, and D. Keim. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 63–72. IEEE, 2012.

[26] E. M. Voorhees et al. The TREC-8 Question Answering Track Report. In *Trec*, vol. 99, pp. 77–82, 1999.

[27] D. Wang, L. Boytsov, J. Araki, A. Patel, J. Gee, Z. Liu, E. Nyberg, and T. Mitamura. CMU Multiple-choice Question Answering System at NTCIR-11 QA-Lab. In *NTCIR*, 2014.

[28] C. Ware. *Information visualization: perception for design*. Elsevier, 2012.

[29] E. Williams. Experimental designs balanced for the estimation of residual effects of treatments. *Australian Journal of Chemistry*, 2(2):149–168, 1949.

[30] A. T. Wilson and K. C. Potter. Toward visual analysis of ensemble data sets. In *Proceedings of the 2009 Workshop on Ultrascale Visualization*, UltraVis '09, pp. 48–53. ACM, 2009. doi: 10.1145/1838544.1838551

[31] R. D. Wolfinger, R. D. Tobias, and J. Sall. Mixed models: a future direction. In *Proceedings of the Sixteenth Annual SAS Users Group Conference, SAS Institute Inc., Cary, NC*, pp. 1380–1388, 1991.

[32] C. Xu, D. Tao, and C. Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

[33] J. Zhao, X. Xie, X. Xu, and S. Sun. Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38:43–54, 2017.

[34] Z. Zheng. Answerbus question answering system. In *Proceedings of the second international conference on Human Language Technology Research*, pp. 399–404. Morgan Kaufmann Publishers Inc., 2002.