PEDANTS: Cheap but Effective and Interpretable Answer Equivalence

Zongxia Li **Ishani Mondal** **Huy Nghiem**

Yijun Liang **Jordan Boyd-Graber**

University of Maryland, College Park

{zli12321, imondal, nghiemh, yliang17, jbg}@cs.umd.edu

Abstract

Question answering (QA) can only make progress if we know if an answer is correct, but current answer correctness (AC) metrics struggle with verbose, free-form answers from large language models (LLMs). There are two challenges with current short-form QA evaluations: a lack of diverse styles of evaluation data and an over-reliance on expensive and slow LLMs. LLM-based scorers correlate better with humans, but this expensive task has only been tested on limited QA datasets. We rectify these issues by providing rubrics and datasets for evaluating machine QA adopted from the Trivia community. We also propose an efficient, and interpretable QA evaluation that is more stable than an exact match and neural methods (BERTScore).[1]

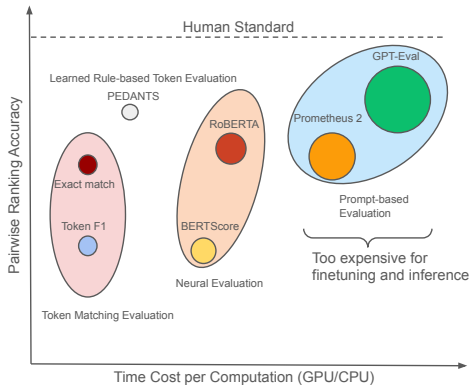

Figure 1: Different evaluation methods have different requirements of computation resources on short-form and factoid QA datasets. Their pairwise ranking accuracies are based on our annotated data in Section 5.3.

1 Introduction

QA evaluation is a necessary pipeline to train and optimize QA models (Sanh et al., 2022). From a model selection perspective, Section 5.3 shows that a weak evaluation can lead to incorrect conclusions about the ranking of models. From a training perspective, having the right objective to train a model is an essential step to guide the model to the right direction Si et al. (2021) shows that keeping models, data, and *test* evaluation fixed, if we improve automatic evaluation during training time, the trained model improves on test set accuracy than using a more rigorous training evaluation– Exact Match (EM).

However, one of the reasons EM is popular is because it is efficient and interpretable compared to other evaluations (Kamalloo et al., 2023). *Twelve out of fourteen* papers (2022-2024) in Table 4 involving QA model training use either EM or F_1 as evaluators while one paper uses GPT-4. Thus, we focus on improving the efficiency, stability, and interpretability of short-form QA evaluation: given a set of gold answers, is the output of a system correct compared to the gold answers? The standard AC evaluations of QA are EM—is the answer string equal to a reference—and token F_1 score—what proportion of tokens match. Neural evaluations such as BERTScore (Zhang et al., 2019) and BERT Matching (Bulian et al., 2022, BEM) calculate pretrained contextual embedding similarity scores between two string inputs. LLMs as a judge such as GPT-4, Claude (Verga et al., 2024), or Prometheus (Kim et al., 2024) use LLMs' reasoning ability and internal knowledge to judge AC.

Standard (EM) and neural evaluations (BEM) are unstable on different styles of QA datasets, where EM is only good for evaluating factoid answers and BEM is mostly good at answers that require reasoning. Section 4 shows that only black-box LLM evaluators (GPT-4) are strong short-form AC judges but they require much more cost and runtime than standard evaluations; open-sourced LLMs struggle at judging incorrect answers, generating many false positives.

To come up with a consistent and structured

[1] https://github.com/zli12321/PEDANTS-LLM-Evaluation

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 9373–9398
November 12-16, 2024

short-form and factoid QA evaluation, we revise existing rules from the Trivia community, integrating standardized AC rubrics from NAQT (National Academic Quiz Tournaments, 2024) and the efficient QA competitions (Min et al., 2021). The rules help us define machine AC, especially for the challenging long tail scenarios (Section 3). We also incorporate difficult QA examples from the *Jeopardy!* community to evaluate current QA evaluation metrics (Appendix G). We then use the our AC framework to prompt GPT-4 (Bubeck et al., 2023) and distil AC knowledge to train a two-level classifier that goes beyond EM and provides a more fine-grained, interpretable and efficient AC evaluation. This broader view of QA evaluation is necessary because modern LLM generate more verbose answers (Zhu et al., 2021), making EM less effective. Out of the examples we analyzed for EM, 90% of answers have correct candidate answers missing from reference answer sets, which humans consider correct but EM does not (Section 5).

Our contributions to automatic AC evaluation are:

1. We adopt AC rules from the Trivia community to build QA evaluation framework.
2. We propose PEDANTS: *Precise Answer Normalizations*. We distil GPT-4 and improve the efficiency and effectiveness of short-form and factoid QA evaluation.
3. We evaluate popular QA datasets and show that only black-box models are good evaluators across eight selected datasets with varying question styles. PEDANTS is more stable and effective than EM and BEM.
4. We analyze hard examples that existing evaluations—token-based evaluations, neural evalations, prompt-based evaluations, PEDANTS—struggle with.

2 Learning Evaluation From the Trivia Community

Using LLMs to evaluate QA is increasingly popular (Chang et al., 2023). However, existing LLM evaluations have inconsistent definitions, rubrics, and prompts to evaluate specific benchmark datasets (Chiang and yi Lee, 2023; Vu et al., 2024). Since LLMs are sensitive to change of prompts/definitions, comparing results among various LLM evaluation papers becomes hard. On the other hand, although EM is still the most popular standard evaluation, its lack of formal correctness definitions makes it unstable and less effective on more verbose QA models (Section 5). The imperfections of current evaluations urges for starting rubrics and frameworks to evaluate models more fairly.

Luckily, we are not starting from scratch! Because many of the QA data and evaluations—from Watson on *Jeopardy!* to TriviaQA—come from the trivia community, Rodriguez and Boyd-Graber (2021) argues that researchers should not just take their data but also examine their *rules* to bridge the gap between NLP and human community needs. Moreover, the Trivia community reflects decades of evaluating AC in high-stakes human competitions that need comprehensive and uniform rules to ensure fairness. Because the subject areas are meant to cover the standard school knowledge, the curriculum breadth reflects many of the evaluation challenges for LLMs. While they are not perfect, they are more comprehensive than any existing automatic QA evaluation framework (Bulian et al., 2022). We propose AC rules from gold standard human QA evaluations: NAQT (National Academic Quiz Tournaments, 2024), *Jeopardy!* (Carberry, 2019), and EfficientQA (Min et al., 2021).

2.1 AC Framework for Evaluation

NAQT is a thirty-year-old purveyor of QA competitions spanning middle schools to "open" competitions that anyone can join. Because of the breadth of participants and judges (often drawn from parent volunteers of participants), the organization has developed easy-to-follow rules that make answer adjudication consistent (otherwise, participants would complain about unfairness). However, NAQT and *Jeopardy!* formats are for *human* players, we cannot unquestioningly adopt the rules for machine QA. For example, some rules cover in-person delivery of answers: mispronouncing koan as cone, pauses between parts, etc. These and other rules are irrelevant to text-based QA.

On the other hand, the rubrics nonetheless provide rules we can adopt. For example, the *specificity* rule is both present in the NAQT and *Jeopardy!* rules, where the responses should be specific enough under the context of a question—*Question: Where is the Eiffel Tower located?*—where the answer *Europe* is incorrect if the given reference answer is *France*, but is acceptable if *Europe* were the intended answer.

Rule	Description
R_1: Entity-aliasing	Widely recognized aliases, pseudonyms that are commonly associated with referred answer entities are acceptable.
R_2: Numerical information	Exact dates, years, numerical values are required unless the question specifically asks for approximations.
R_3: Less details	The answer provides less detail but should include the essential and correct information required by the question (specificity level 1).
R_4: More details	The answer contains additional information that does not contradict the question or initial answer (specificity level 2).
R_5: Semantic equivalence	A high degree of word overlap does not establish equivalence. The answer must be contextually and semantically accurate.
R_6: Irrelevant information	Any irrelevant or inaccurate description related to the question should be considered incorrect.
R_7: Other possible answers	The response is correct but not in the initially provided list.

Table 1: The correctness of a candidate answer can be traced and categorized to one or more of the above rules. We adopt the rules from NAQT modified after analysis of an annotated dataset with human correctness judgments between reference answers and machine generated answers. The acceptability of QA model answers are based on the rubrics. Rules with examples are in Table 4 (Appendix).

The revising process To find the rules that apply to text-based AC, We manually annotate 200 examples from the AC test (Bulian et al., 2022) set where BEM and human judgments misalign. We then find the rules from the trivia rubrics that explain the disagreement and revise them to remove human-specific desiderata (Table 1 with QA examples in Appendix A).

3 Efficient QA Evaluation

Evaluation is not a task like QA itself where the size of the model is a secondary consideration; an effective method (i.e., one that could be adopted) needs to be accurate, fast, and low-cost (Rieger and Hansen, 2020). This section's goal is to match the effectiveness of the LLM evaluation while minimizing computation, latency, and disk space. EM and token F_1 evaluations are fast and require no storage but are superficial. LLM evaluations are trainable but require substantial disk space, cost, computational time, and latency. In aggregate, they might be too expensive for universal adoption in standard QA training and evaluation pipelines.

By revising AC for text-based QA, we can construct an efficient automatic evaluation that not only aligns with our AC decision-making process but also goes beyond token-level matching that follows the human mental judgment process.

• q_1: Who is the president of the US in 2023? • a_1: Joe Biden • $\tilde{a}_1$: Joseph Biden • q_1 is asking about *who*, having a question type T_{who}. • $(q_1, a_1, \tilde{a}_1)$ is classified as Rule R_1-commonly known entity aliases are equal • Token F_1 : 0.5, precision 0.5, recall 0.5. • Under the constraint of R_1 and T_{who} $\rightarrow$ **correct**
• q_2: When did Joe Biden become the president of the US? • a_2: Jan 20, 2021 • $\tilde{a}_2$: 2021 • q_1 is asking about *when*, having a question type T_{when}. • $(q_2, a_2, \tilde{a}_2)$ is about numerical dates and years–R_2 • Token F_2: 0.5, precision 0.33, recall 1.0. • Dates and years should have an exact match $\rightarrow$ **incorrect**

Table 2: Despite identical token F_1 scores for the two QA examples, the judgment decisions vary due to different question types and evaluation rules.

3.1 PEDANTS Details

This section details how we train the PEDANTS pipeline. This is a two-step classifier that first predicts the type T of the question and which of the rules R from Table 1 are applicable. It then creates a final decision of whether given a question q, a reference answer a, and a candidate answer $\tilde{a}$, PEDANTS classifier decides whether a candidate answer is correct or not. We first define what question types and AC rules, describe how we train those preliminary classifiers, and then describe the training of the final classifier.

Question Type Let T denotes the type of a question, categorized by {*who*, *why*, *how*, *what*, *when*, *where*, *which*}. Specifically, the type of a question q primarily depends on the content of q itself, but it may also depend on the context of reference answer a. Therefore, we define the type T of the pair (q, a) as one of the type categorizations.

Applicable Rules Given pair (q, a, $\tilde{a}$), R_n is the rule from Table 1 used to judge the correctness of $\tilde{a}$. For example, if the question is asking about *Joseph Robinette Biden*, then at a minimum the answer must contain *Biden* (the family name), but *Robinette* would not be enough.

Both of these are necessary prerequisites before we can make a final AC decision. This is because *which* and *how many* tokens must overlap between reference and candidate answers depends on the type of questions and applicable rules.

Mirroring human judgments Table 2 shows that the necessary token overlap for determining AC varies with the question type T and rule R_n, mirroring the variability in human judgment. In this context, we refer to T and R_n as our question type features and rule features. An automated answer AC judgment needs to balance flexibility with specific constraints to ensure accurate judgments on various edge cases:

1. Person names and entities may match despite low token overlapping scores, as variations in aliases (e.g. *Joe* vs. *Joseph*) can obscure lexical likeness.
2. If the answer includes extra details but the core information is correct, we need a lower token matching threshold and high recall to ensure the essential information is answered (e.g. *France* vs. *France, Europe*).
3. For date and numerical answers, stricter token matching is required, while type features (T) becomes important for identifying numerical answers, etc.

In the evaluation of $(q_1, a_1, \tilde{a_1})$, the feature R_1 and T_{who} allow for correctness judgments even when the candidate answer $\tilde{a_1}$ and the reference answer a_1 do not exactly match or share high token F_1, precision, or recall scores. Conversely, exactness in dates is required for $(q_2, a_2, \tilde{a_2})$ categorized under feature T_{when} and R_2. Hence, even though the token F_1 match between a_2 and $\tilde{a_2}$ is similar to that of a_1 and $\tilde{a_1}$, the evaluation penalizes the lack of exactness in date answers (Table 2). The combination of token F_1, precision, and recall thresholds is tailored to the specific question type and rule. Therefore, features such as T, R, TF-IDF encodings, and *standard metric thresholds can be learned from data* that is representative of the AC rules.

Training Data Preparation We manually write five to ten representative QA examples with human judgments as seed examples. Then we prompt GPT-4 with the seed examples that contain type T and rule R_n and ask it to self-verify its generated judgment, T, and R_n assignment (Ren et al., 2023).[2] In the AC training data, each example is paired with $(q, a, \tilde{a}, T, R_n)$, where T and R_n are categorical variables initially, but are treated as distribution of predictions when training the feature extraction classifiers. We then prompt GPT-4 to assign T and R_n to all examples in Bulian et al. (2022)'s AC training set (9,090 examples). We merged our generated data with Bulian et al. (2022)'s examples as PEDANTS' training set (exact details and prompt templates in Appendix B).

Training PEDANTS We train two feature extractors, $F(R)$ and $F(T)$, to extract features for rule and question type features. For each pair $(q, a, \tilde{a})$ in the training set, we lemmatize, remove punctuation, and calculate token F_1, precision, and recall. We encode $(q, a, \tilde{a})$ as *[CLS] q [SEP] a [SEP]* $\tilde{a}$ using TF-IDF, and concatenate the outputs from $F(R)$, $F(T)$, and the token scores to train a logistic regression classifier. Human annotations or GPT-4 judgments are used as AC labels. For example, if q is *Who is the president of the US in 2023?*, with a as *Joe Biden* and $\tilde{a}$ as *Joseph Biden*, $F(R)$ classifies the rule as R_1 (entity aliasing), and the question type as T_{who}, indicating a person. Feature extractors are also logistic regression classifiers. Details on classifier hyperparameters are provided in Appendix 3.

In sum, PEDANTS is a pipeline classifier, with a feature extraction stage where TF-IDF encoded (q, a) and $(q, a, \tilde{a})$ are processed to extract rule R and type T features. This is followed by a decision-making stage where T and R features, along with token F_1, precision, and recall and TF-IDF encodings, are input into a final logistic regression classifier to predict AC. PEDANTS's architecture simulates human judgment by integrating linguistic and rule-based features.

4 Evaluation Setup

An effective evaluation of evaluation methods requires validations on diverse data and models. We want to test limitations automatic evaluations on many datasets and QA models to 1: verify their effectiveness on different styles of datasets, evaluating its strengths over other metrics and weaknesses regarding different the styles of answers; 2: evaluate their stabilization across answers generated by various QA models (different sizes, different pretraining data, black-box and open-source models...) on the same dataset, which is important to provide a more fair comparison to QA models on a benchmark; 3: ensure the test data is not restricted by the patterns of the training data to prevent overfitting on learned metrics.

[2] Asking GPT-4 to self-verify its generation can improve the quality of its generations. Prompt Templates in Appendix Table 6.

Dataset selection We select three benchmark datasets that have factoid and short-form answers, four benchmark data that have answers in phrases or sentences (not summarization), and one with human generated answers. The three factoid QA datasets are sampled from NQ-OPEN (Kwiatkowski et al., 2019), NARRATIVEQA (Kočiský et al., 2018), and HotpotQA (Yang et al., 2018). These three datasets are standard QA benchmarks, used to train many QA models. The most popular metrics for evaluating the models on these datasets are EM and token F_1 score. We use these datasets to verify that PEDANTS is at least as good as EM and BERTScore on short-form QA datasets.

We also sample from *Biomedical Machine Reading Comprehension* (Stavropoulos et al., 2020, BIOMRC), *Microsoft Machine Reading Comprehension* (Bajaj et al., 2018, MS MARCO), *Conversational Question Answering Challenge* (Reddy et al., 2019, COQA), and *MOdeling Correctness with Human Annotations* (Chen et al., 2020, MOCHA) to test more challenging evaluation, where the answers are sentences that can confound EM.[3]

Furthermore, we collect a dataset from *Jeopardy!* , featuring answers from real human players. This dataset challenges existing evaluation metrics such as EM, which by design scores 0 in Macro F_1 evaluations. Evaluating these answers correctly is hard and requires years of human experience.[4]

Model Selection Different QA models pretrained with different data generate different answers, and their generations can also depend on model sizes. We select a range of models to generate answers for selected datasets Flan-T5-xl (3B) (Chung et al., 2022), LLaMA 2 (7B) (Touvron et al., 2023), GPT-3.5-TURBO (a black-box model), Mistral 8x7b Instruct (Jiang et al., 2024), Yi-Chat 34B (AI et al., 2024), and LLaMA 2 70B (Touvron et al., 2023). Our models span 3B to 70B include both black-box and open-sourced architectures, and incorporates models pretrained with different sources.

Evaluation metric selections We choose evaluations from three mainstream methods to compare with PEDANTS: token evaluation, neural evaluation, prompt-based evaluation. Specifically, token evaluations are EM and token F_1, which compare the similarity of two strings; neural evaluations are BERTScore (Zhang et al., 2019), and ROBERTa matching (Bulian et al., 2022), which use pretrained or finetuned transformer embeddings to measure the similarity between two strings; prompt-based evaluations are GPT-4 and Prometheus-2 (Kim et al., 2024), which use finetuned generative models to generate an evaluation score.[5] PEDANTS uses question and answer types to judge answer correctness with learned optimal string similarity matching.

4.1 Annotated Dataset Details

Since MOCHA is a benchmark evaluation dataset, it includes answers generated by GPT-2 and human rating annotations (Likert scale 1-5). We treat an answer as correct if it meets a threshold score of 4.

The *Jeopardy!* data is a real-world QA dataset with expert correctness judgements. Each example in the dataset has a question, a gold answer, a response by *Jeopardy*! players, and expert judgment of the response. The dataset includes candidate answers that are challenging for experts to judge. For example, given a question *Your surgeon could choose to take a look inside you with this type of fiber optic instrument*, the gold answer is *laparoscope*, but *endoscope* was given and ruled incorrect during the show that later reverted to correct, verified by professionals in accordance with the *Jeopardy*! answer acceptability rules. The *Jeopardy!* dataset is meant to challenge both human judges and automatic evaluations to show the imperfections of automatic evaluations. We collect 504 examples with examples in Appendix G.

4.2 Annotating Generated Data

Only MOCHA and *Jeopardy!* have available answers and judgements, so we use GPT-3.5-TURBO, Flan-T5 xl, and LLaMA 2 7B to generate answers for the three factoid and short-form datasets–NQ-OPEN, HotpotQA, and NQA; we also use Mistral 8x7b, Yi-Chat 34B, and LLaMA 2 70B to generate answers for challenging datasets–COQA, BIOMRC, and MS MARCO. We filter out examples where the reference and candidate answers do not have an exact match. The total number of filtered examples is 42,090. We then hire annotators on Prolific whose first language is English, with at least a community college degree completed and above 99% approval

[3] We show failure modes of token matching for evaluation and that PEDANTS is more robust than neural methods like BERTScore for datasets with longer gold answers.

[4] https://www.j-archive.com. More details of these evaluation sets, including the number of examples, example questions and answers are provided in Table 5 and Appendix G.

[5] Prompt templates for GPT-4 are in Appendix Table 8

rate to judge the correctness of the filtered examples using our AC rubrics in Table 1. Out of 42,090 examples, 6,626 examples have annotations from two annotators, and the interannotator agreement of the 6,626 examples has Krippendorff's $\alpha = 0.75$. For the examples that have two annotations where annotators disagree, we have the authors annotate them to break tie the judgments.

5 Error Analysis

With available datasets with human judgments, this section analyzes the human agreement, time and monetary cost, and model ranking accuracy. Then we dive into examples that are challenging to the standard evaluation methods (token matching and neural evaluation) but can be handled by PEDANTS.

5.1 Human Agreement Rates

For each dataset, we calculate each selected metric's accuracy and Macro F_1 score between automatic judgements and human judgments. EM and token F_1 have high human agreement on short QA types—NQ-OPEN and Hotpot QA—but falters for more challenging QA datasets (Figure 2). BERTScore can handle the more challenging datasets that have longer gold answers but falls behind token matching on short-QA. ROBERTa, PEDANTS, and GPT-4 have similar and stable human agreement. PEDANTS is better than ROBERTa for short-form QA datasets, but ROBERTa is better for long-form datasets that require a semantic understanding of word relationships.

Since Prometheus 2 is finetuned on evaluation data with long reference answers, it shows low human agreement on short-form QA datasets by frequently misjudging simple cases such as the candidate *No.* *versus* the gold answer *No* as incorrect. However, Prometheus has higher human agreements than EM on more challenging long-form QA datasets, which shows that prompt-based evaluations are not stable across different datasets. Therefore, a more expensive method is not always better than a cheap method.

5.2 Prompt-Based Evaluation Are Costly

Token and neural evaluations are cheap but are inflexible. Prompt-based evaluation like GPT-4 are the best if ignoring costs. However, computation cost and time are important in real life practices, particularly if evaluation is a step during QA model training. Prompt-based evaluations are much more expensive than other evaluations regarding runtime and monetary costs (Table 3). Open-source models like Prometheus require extensive GPU resources. Since model training is already a GPU intensive and time-consuming task, an extra GPU for evaluation might not always be available or possible. Although API-based services like GPT-4 require no GPU, its generations often change with periodic updates (Brożek et al., 2023), leading to inconsistent evaluations. In addition, GPT-4 API calls have high latency, requires network connection, and banned in fifteen countries as of 2024 (Click4Assistance, 2023), which is expensive and not accessible for every researcher to reproduce results. In a realistic case, suppose a user who has 10,000 training data tries to train three models and pick out the best one, for a three epoch training, the total GPT-4 evaluation time is 21 hours and costs *a thousand* dollars and approximately 7 days using Prometheus on an A6000 GPU. On the other hand, one of the reason that EM or token F_1 are popular is that they are fast and easy to implement with minimum costs, but they are also less effective than prompt-based evaluations and PEDANTS.

Metric	Runtime/(min)	Disk Space	Cost
EM	0.05	0	0
Token F_1	0.05	0	0
PEDANTS	1	5 MB	0
BERTScore	4.95	499 M	0
ROBERTa	4.95	499 M	0
Prometheus 2	1,123	12 GB	0
GPT-Eval	140	unknown	$120

Table 3: Though GPT-4 is the best evaluation model, it costs more money and time. The runtime is based on per 10,000 examples from 8 test datasets on 7 QA model. BERTScore, LERC, ROBERTa, and prometheus 2 evaluation methods are run on a single A6000 GPU device, and other methods are run with local CPU. The API cost for GPT-4-Eval is about $120.

5.3 Pairwise Ranking Accuracy

Model ranking is one of the most important use of evaluation. Human evaluations are expensive and time consuming, and they are usually used for big development cycles. However, automatic evaluations are usually used during development cycles. Thus, knowing which version of model is better on a task is essential to guide the model into the right development direction and prepare better training and evaluation data.

For each dataset with responses from N models, we define pairwise ranking (PWR) accuracy as the percentage of model pairs that an automatic metric

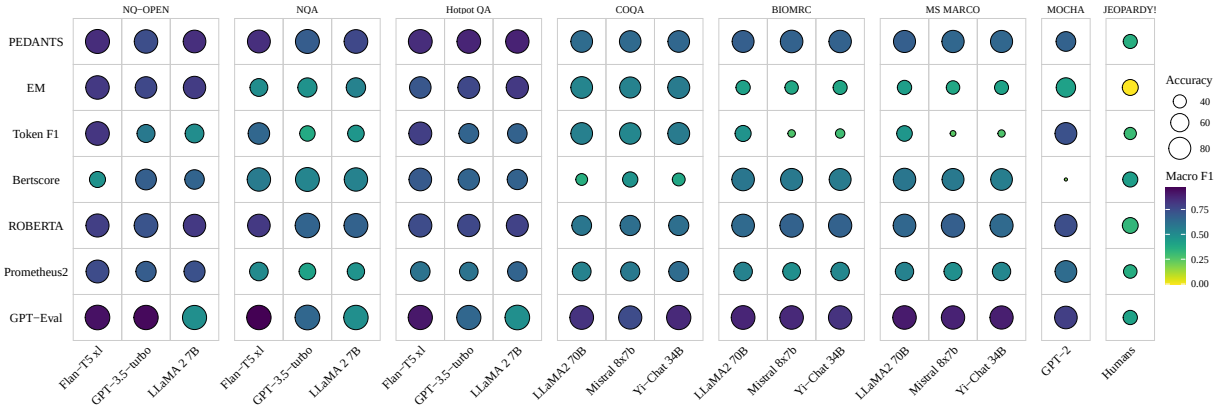

Figure 2: The size of the circles shows each metrics' human agreement accuracy and the color shows the Macro F_1 score . We put PEDANTS first for ease of visualization. EM, token F_1, and BERTScore have unstable human agreement on different QA datasets by looking at horizontal circle size variations– ranging from 25% to 90%. ROBERTa, PEDANTS, and GPT4-Eval are more robust and stable with varying QA datasets. Although Prometheus 2 is fine-tuned for evaluation purposes, it fails on short-form QA. PEDANTS is less costly than GPT-4, Prometheus 2, ROBERTa, and BERTScore and has more stable human agreements across seven evaluation datasets than EM. Prometheus assigns scores from 1 to 5; we use 4 or higher as indicative of correctness.

have the same ranking as human rankings:

$$\frac{1}{\binom{N}{2}} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} [(\text{rank}_{\text{metric}}(i,j) = \text{rank}_{\text{human}}(i,j))] \quad (1)$$

Where $(\text{rank}_{\text{metric}}(i,j) = \text{rank}_{\text{human}}(i,j)$ is 0 if automatic metric disagrees with human rankings, and 1 otherwise.

We calculate each metric's pairwise ranking accuracy against human rankings for datasets that have responses from multiple models. GPT-Eval and PEDANTS are more stable at ranking models correctly on simple and more challenging QA datasets. PEDANTS ranks all models correctly on four datasets (Figure 3).[6] Surprisingly, while BERTScore show inconsistent rankings on more challenging datasets, it has higher human agreements than EM and Prometheus (Figure 2). From answer-level comparison (human agreements) and model-level comparison (pairwise ranking) of evaluation metrics, GPT-Eval is the most robust and recommended evaluation, followed by PEDANTS.

5.4 Error Analysis

We analyze failures of token, neural, prompt-based, and PEDANTS evaluations by manually analyzing 441 randomly sampled examples where they disagree with human judgments. We categorize errors where these methods are likely to fail. Then we examine what features contribute PEDANTS' most correctness judgments.

[6]We use the percentage of responses humans or a metrics judge as *correct* on a dataset to assign a score (between 0 to 100%) to a model, then rank models with descending scores.

Figure 3: The pairwise ranking accuracy for dataset that have multiple model responses. The *TOTAL* is the pairwise ranking accuracy across all six datasets. PEDANTS and GPT-Eval rank more models correctly than other methods on most datasets.

5.5 Failures of Token and Neural Evaluations

We randomly select 63 examples from our annotated dataset where the methods—EM, token F_1, BERTScore, and ROBERTa, Prometheus 2, GPT-Eval, and PEDANTS——show disagreement with human judgments. Two annotators manually analyze the 441 examples. We discuss the failure cases separately within the contexts of token-level, neural, prompt-based, and PEDANTS evaluations.

Token evaluations are inflexible Token evaluations makes more sense for *extractive* QA, where the goal is to find answers embedded in a fixed passage and compare token-level similarities of two strings. However, generative models are capable of generating many valid and correct answers, making token evaluations less effective. About 90% (56) of the examples considered correct under human judgment are incorrect using EM. EM and token F_1 evaluations make the most mistakes on the following categories:

1. *Semantic similarity*: token evaluations struggle with catching semantic equivalence between two strings. For example, *reference: By labeling it satire [SEP] candidate: as being satirical.* both convey the same meaning but EM and token F_1 will never judge them equivalent.
2. *Specificity*: this falls under AC Rule 4, where LLMs often generate more detailed answers than gold answers. A representative example where EM and token F_1 both fail is *question: The Aviation Hall of Fame and Museum of New Jersey is located at the airport in which New Jersey county? [SEP] reference: Bergen [SEP] candidate: Bergen county, New Jersey*.

Neural evaluations cannot make sharp distinctions Unlike token-based methods, BERTScore and RoBERTa matching leverage pretrained or finetuned AC knowledge and neural model representations, using BERT (Devlin et al., 2018) to compute cosine similarity between the embeddings of the reference and candidate answers for evaluating correctness. Because RoBERTa matching is finetuned specifically for AC task, it gives a more fine-grained evaluation than BERTScore. However, one weakness of neural evaluation, even with finetuning, is its difficulty in distinguishing terms that are similar in the embedding space but have different meanings. This weakness is especially obvious when the reference and candidate answers are short, where pretrained biases skew judgments. Many of the incorrect answers that BERTScore and RoBERTa give high similarity scores and judge as equivalent are terms such as *apple and pear; king and queen; left and right; item A and item B; etc*).

Prompt-based evaluation decisions are unstable Black box models and open-source LLMs have very different weaknesses. Although black box models such as GPT-4 are strong evaluators, it is not sensitive to *Specificity* due to word tokenizations during pretraining stage. GPT-4 occasionally misjudges regarding the specificity of dates or locations. For example, *Central Germany and Western Europe* are judged equivalent by GPT-4, where Western Europe can refer to many countries but Central Germany is only one country. *Reference: March 25, 2018 and candidate: 2018* are also considered equivalent by GPT-4 when the question is asking for *when* and the expected answer is expecting a specific date.

On the other hand, Prometheus is finetuned with mass synthetic evaluation data to evaluate the quality of long generations instead of just AC, but it often misjudges simple short answers that are obviously wrong compared with the reference answers. Prometheus often assigns a low rating when differences between the reference and candidate answers involve missing symbols, punctuation, or unnecessary omissions, such as *reference: four albums* and *candidate: four*. In this case, the question is *How many albums had been recorded by Talking Heads by November, 1980?*, but Prometheus fails to consider the context of the question in its evaluation. Prometheus also struggles at simple cases like whether *Yes* and *No* as equivalent, indicating that open-source LLMs may outperform standard evaluations like EM or BERTScore in evaluating long answers but are less effective at evaluating short-form factoid answers.

PEDANTS lacks world knowledge and commonsense reasoning Although PEDANTS uses learned question types and AC rules to judge AC that goes beyond simple string matching, it is still a fairly cheap classifier with limited vocabularies. From the 63 examples we analyzed, PEDANTS cannot handle complet word relationships but can be improved with future work. *Synonyms that are not in the training vocabulary*: Unlike pretrained transformer models, PEDANTS lacks the general world knowledge to understand relationships between similar words. If the reference or candidate answer contains vocabularies that PEDANTS does not recognize such as *cold and chilly*, PEDANTS becomes similar to token-based evaluation, which is a trade off for smaller model sizes, faster runtime, and less pretraining biases.

Commonsense reasoning and fact-checking remain challenging tasks for all QA evaluation We analyze 45 examples from *Jeopardy!* where

all methods disagree with expert judgments. The 45 examples require more complex commonsense reasoning and fact-checking by human experts, which significantly challenge current QA evaluation methods. An interesting and challenging example is *Question: Noted anarchist Prince Peter Alexeivitch Kropotkin wrote a 19th century entry on this capital? Reference: Moscow Candidate: Saint Petersburg*. The answer was also overruled to be correct by the panel with the fact that Kropotkin lived from 1842 to 1921; during his lifetime, the capital of *Russia* was changed from *Saint Pertersburg* (1712-1918) to *Moscow* (1918-). Thus, the capital Kropotkin referred to could be either of the answer. Experts assess correctness based on social norms, reasoning, and word choice, a process far more complex than mere string comparison.[7] Evaluating a hard answer requires validating a fact and reasoning over a fact, if we want to adopt data from the expert community for evaluation, training, or analysis, we need to respect the community and also adopt their rules to improve QA evaluations.

PEDANTS Feature Importance Understanding which features enable PEDANTS to make accurate judgment decisions can help us interpret its evaluation process and improve model transparency. We use coefficient interpretation (Hosmer and Lemeshow, 2000) to identify the top five features that significantly influence PEDANTS's decision-making: R_1, F_1 score, T_{what}, R_3, and tf-idf. These features show the critical role of AC rules, question types, and F_1 scores as features in enhancing PEDANTS's judgment accuracy, showing that a well-defined rules framework is important for PEDANTS to outperform traditional metrics like EM, F_1, and neural evaluation methods on our tests.

6 Related Work

Drawing on community expertise There is a recent trend of using data from well-defined communities ethically (Shilton, 2016; Rogers et al., 2021), and the trivia community is no different. While their data have been used for NLP tasks—from Watson's *tour-de-force* on *Jeopardy!* to TriviaQA—the data have been used without attending to the norms of how the trivia community uses it. We should listen to their evaluations if we adopt their data.

Machine QA evaluations QA evaluations span a continuum—easy to assess using EM for short answers, yet challenging for long-form QA, which have many valid answers (Xu et al., 2023). However, even short-form QA can be challenging when it requires contextual understanding or have huge combination of valid output spaces(Bulian et al., 2022). Such tasks can reach NLP-completeness, intersecting diverse NLP domains like coreference, translation, and entailment.

With the continual development of machine QA models' ability to answer questions, the area of QA evaluation still falls behind, where most research papers still use standard evaluation metrics–EM or F_1 (Table 4). Chen et al. (2019); Kamalloo et al. (2023) both point out pitfalls of current standard evaluation metrics, where QA is not merely extracting exact answers from source texts in the era of large generative models. Bulian et al. (2022); Kamalloo et al. (2023) validate fine-tuned BERT on human annotated AC datasets is better than standard metrics and closer to human judgments. Wang et al. (2023) demonstrates that BERT-based evaluation methods like BERTScore and BLURT (Sellam et al., 2020) are no better than standard metrics, with dataset sensitivity to BERTScore threshold settings and unclear boundary definitions across datasets, while generative LLM evaluators like GPT-3.5 tend to make *over-generalization* errors that BERT-based methods and humans typically avoid. Yet the ability of generative LLM evaluators is not evaluated with clear definitions of AC rigorously.

7 Conclusion

Automated QA evaluation is an important pipeline for developing better models. However, evaluation metrics like EM are popular for their efficiency, but the lack of AC rubrics and diverse evaluation datasets still hold us from exploring and building more robust and stable QA metrics competitive with LLM evaluations, which is not always available. A basic beginning framework derived from QA human experts can enhance and generalize automated evaluation methods' agreement more with human judgments. Though not perfect, PEDANTS requires few computational resources and time but more robust across benchmark datasets. Future work can integrate and refine these rubrics into long-form QA and novel QA, where we can collect better evaluation data and improve fact-checking and commonsense reasoning of AC; future work can also combine efficient metric with LLM to improve runtime evaluation efficiency and reduce cost.

[7] See examples in Appendix J

Limitations

Since we are the first to adopt and revise answer correctness rules from the Trivia community, they are still not the perfect within the machine evaluation paradigm. We advocate stronger connections between the computer linguistic community and the professional Trivia QA community to improve the interpretability, efficiency, and effectiveness of rule-based QA evaluation. Furthermore, although PEDANTS has shown its efficiency and effectiveness on standard QA benchmark datasets than token evaluation and neural evaluations, it still has drawbacks that need to be address. The effectiveness of PEDANTS is constrained by its limited vocabulary. When encountering words outside its training corpus, PEDANTS becomes to the less-effective token-level evaluation. This vocabulary limitation also impedes PEDANTS's capacity to capture complex word associations, thereby weakening its commonsense reasoning abilities. Future research is needed for expanding PEDANTS's vocabulary range and improve its ability to learn complex relationships between words. Such improvements can bridge the gap between efficient PEDANTS evaluation and expert human evaluations.

Ethics

Our annotation process prioritizes participants' privacy. We do not collect any personal information from participants, and they are free to exit the study within ten minutes of starting. The Institutional Review Board (IRB) has reviewed and exempted our annotation protocol. Annotators are compensated at a rate of $15 per hour, with an expected completion of 300 examples in this time frame. For productivity beyond 300 examples, we offer a bonus of 2.5 cents per additional example. Our open-source QA evaluation Python package is released under the MIT license, encouraging public use and contribution. We welcome its application for any purpose, supporting transparency and advancement in the field.

Acknowledgement

We thank the invaluable contributions from University of Maryland CLIP members who initiated this project through their insightful brainstorming. We extend our gratitude to the annotators for their efforts in annotating the data and for Jonah Greenthal for his discussions of answer equivalence. Additionally, we appreciate the anonymous reviewers for their constructive feedback, which has significantly enhanced the robustness and rigor of our QA evaluation and analysis. The improvements in PEDANTS owe much to their valuable comments and suggestions. This material is based upon work supported by the National Science Foundation under Grant No. IIS-2403436 (Boyd-Graber). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints.

Haozhe An, Zongxia Li, Jieyu Zhao, and Rachel Rudinger. 2023. Sodapop: Open-ended discovery of social biases in social commonsense reasoning models.

Haozhe An and Rachel Rudinger. 2023. Nichelle and nancy: The influence of demographic attributes and tokenization length on first name biases.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset.

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen tau Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers.

Bartosz Brożek, Michał Furman, Marek Jakubiec, and Bartłomiej Kucharzyk. 2023. The black box problem revisited. real and imaginary challenges for automated legal decision making. *Artificial Intelligence and Law*, 32:1–14.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar,

Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Boerschinger, and Tal Schuster. 2022. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. *arXiv preprint arXiv:2202.07654*.

Matt Carberry. 2019. Jeopardy! casebook. http://jeopardy.mattcarberry.com/casebook.html. Accessed: 2024-01-12.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2023. A survey on evaluation of large language models.

Allen Chen and Okan Tanrikulu. 2024. Improving qa model performance with cartographic inoculation.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124, Hong Kong, China. Association for Computational Linguistics.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2020. Mocha: A dataset for training and evaluating generative reading comprehension metrics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Cheng-Han Chiang and Hung yi Lee. 2023. A closer look into automatic evaluation using large language models.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge.

Click4Assistance. 2023. What countries is chatgpt banned in and how this will impact live chat for businesses. Accessed: 2023-04-13.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

David W. Hosmer and Stanley Lemeshow. 2000. *Applied Logistic Regression*, 2nd edition. Wiley Series in Probability and Statistics. Wiley.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. Association for Computational Linguistics.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti,

Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Chatqa: Surpassing gpt-4 on conversational qa and rag.

Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. Sail: Search-augmented instruction learning.

Meta Platforms, Inc. 2024. Llama-3.1-8b-evals. `https://llama.meta.com/llama-downloads`. Llama 3.1 Version Release Date: July 23, 2024.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Sewon Min, Jordan Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, Tom Kwiatkowski, Patrick Lewis, Yuxiang Wu, Heinrich Küttler, Linqing Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Edouard Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, Shun Sato, Ryo Takahashi, Jun Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, Pavel Smrz, Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen-tau Yih. 2021. Neurips 2020 efficientqa competition: Systems, analyses and lessons learned. In *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, volume 133 of *Proceedings of Machine Learning Research*, pages 86–111. PMLR.

National Academic Quiz Tournaments. 2024. Correctness guidelines. Accessed: 2024-01-12.

Huy Nghiem, John Prindle, Jieyu Zhao, and Hal Daumé III. 2024. " you gotta be a doctor, lin": An investigation of name-based bias of large language models in employment recommendations. *arXiv preprint arXiv:2406.12232*.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge.

Jie Ren, Yao Zhao, Tu Vu, Peter J. Liu, and Balaji Lakshminarayanan. 2023. Self-evaluation improves selective generation in large language models.

Laura Rieger and Lars Kai Hansen. 2020. Irof: a low resource evaluation metric for explanation methods.

Pedro Rodriguez and Jordan Boyd-Graber. 2021. Evaluation paradigms in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9630–9642, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Anna Rogers, Timothy Baldwin, and Kobi Leins. 2021. 'just what do you think you're doing, dave?' a checklist for responsible data use in NLP. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4821–4833, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Katie Shilton. 2016. Emerging ethics norms in social media research. *Big Data Ethics*.

Chenglei Si, Chen Zhao, and Jordan Boyd-Graber. 2021. What's in a name? answer equivalence for open-domain question answering.

Petros Stavropoulos, Dimitris Pappas, Ion Androutsopoulos, and Ryan McDonald. 2020. Biomrc: A dataset for biomedical machine reading comprehension.

Joseph Stone and Tim Yohn. 1992. *Prime Time and Misdemeanors: Investigating the 1950s TV Quiz Scandal: A D.A.'s Account*. Rutgers University Press, New Brunswick, N.J.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense

knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

National Academic Quiz Tournaments. 2019. Why does ken jennings play quiz bowl? Online Video. URL: https://www.youtube.com/watch?v=Y5ku181Zm8I.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating llm generations with a panel of diverse models.

Tu Vu, Kalpesh Krishna, Salaheddin Alzubi, Chris Tar, Manaal Faruqui, and Yun-Hsuan Sung. 2024. Foundational autoraters: Taming large language models for better automatic evaluation.

Cunxiang Wang, Sirui Cheng, Qipeng Guo, Yuanhao Yue, Bowen Ding, Zhikun Xu, Yidong Wang, Xiangkun Hu, Zheng Zhang, and Yue Zhang. 2023. Evaluating open-qa evaluation.

Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel. 2019. Nlprolog: Reasoning with weak unification for question answering in natural language.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963, Online. Association for Computational Linguistics.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023. A critical evaluation of evaluations for long-form question answering. *ArXiv*, abs/2305.18201.

Rongwu Xu, Xuan Qi, Zehan Qi, Wei Xu, and Zhijiang Guo. 2024. Debateqa: Evaluating question answering on debatable knowledge.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Gal Yona, Roee Aharoni, and Mor Geva. 2024. Narrowing the knowledge evaluation gap: Open-domain question answering with multi-granularity answers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6737–6751, Bangkok, Thailand. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Merging generated and retrieved knowledge for open-domain qa.

Yuhang Zhou and Wei Ai. 2024. Teaching-assistant-in-the-loop: Improving knowledge distillation from imperfect teacher models in low-budget scenarios.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

Wang Zhu, Jesse Thomason, and Robin Jia. 2023. Chain-of-questions training with latent answers for robust multistep question answering.

A AC and Correctness Guideline Examples

Table 4 presents additional typical QA example pairs, along with corresponding judgments and explanations. These examples provide clarity on how rules in Table 1 are applied.

B Training Data Augmentation

Since no existing AC datasets are built according to our rules, we will generate additional synthetic data and augment the existing AC dataset from Bulian

Question	Reference	Candidate	Judgment
Who plays Red on Orange is the New Black?	Kate Mulgrew	Katherine Maria Mulgrew	Correct aliasing name
Wher can I get a state issued id in US?	DMV	Department of Mobile Vehicles	Incorrect entity under rule 1
What year did World War II end?	1945	Sep 2, 1945	Correct under rule 2 and 4. Exact date and more information
How tall can a giraff grow?	16-20 feet	18 feet	Incorrect under rule 2: answers must be a range
Who discovered pennicillin?	Scottish physician Alexander Fleming	Alexander Felimng	Correct under rule 3: less details
What followed the late local programming after Super Bowl 50?	The Late Late Show with James Corden	The Late Show	Incorrect under rule 3
When did Morales launch his policy in the eastern lowlands?	2009	Augst 3, 2009	Correct under rule 4: more detailes
Protective coloring is common in what insect family?	Beetle	Bettle and Formicidae	Incorrect additinal information
What is the primary source of energy for the Earth?	Solar radiation from sun	Solar energy from sun	Correct under rule 4
Which group has higher likelyhood of brain damage?	group A	group B	Incorrect under rule 4
When did the Golden State Warriors win the NBA championship?	2015, 2017	2022	Correct under rule 7
When does Pam find Jim dead?	In a bathtub while in Paris	Paris, France	Incorrect under rule 4: not enough specificity
What flattens Aron's hand and wrist?	A boulder	A rock	Incorrect under rule 5, a boulder is a very large rock but a rock can be any size
Based on March 1937, this man was also a citizen of the United Kindom?	Alistair Grant	Sir Matthew Alistair Grant	Correct under rule 5: common aliases
What is the issue type of the inner most layers of cells?	Epithelium	Epithelial tissue	Correct under context
Are Nikilaschka and White Russia made of beer?	No	No, Nikolaschka and White Russian not made with beer	Correct under rule 5
Reggaetón Lento is a song by the boy band formed on which date?	Dec 13, 2015	Dec 15, 2015	Incorrect based on rule 2 and 4: Date and numerical should be exact
How are Tessa and Tito connected?	They are secrectly married and have two children	Megan	Incorrect under rule 7: irrelevant answer
What is the capital of France?	The capital of France is Paris	Paris is the largest city in France	Incorrect under rule 6: high token overlap but referring to different information
Your surgeon could choose to take a look inside you with this type of fiber optic instrument?	laparoscope	endoscope	Correct under rule 1 and 4: "endoscope" encompasses a variety of instruments used for viewing different internal parts of the body, not limited to the abdominal or pelvic area
I've seen the promised land, I may not get there with you. but...we, as a people, will get to the promised land	Martin Luther King	Martin Luther King in "I had a dream"	Incorrect under rule 5, the question is not from the speech
Sarah bought a dress and a hat. Which item was more expensive?	The dress cost more than the hat	The item Sarah spent less on was the hat	Correct under rule 6 with semantic equivalence

Figure 4: We list out more relevant AC pairs with judgments under our revised rules. Some of the candidate responses are manually written, some of them are from *Jeopardy!* , and some of them are generated by various QA models such as Flan-t5.

et al. (2022). By combining this synthetic data with real human-annotated examples, we aim to enhance PEDANTS' performance by distilling knowledge from GPT-4 (Zhou and Ai, 2024).

We first build representative QA example pairs that align with the revised AC and acceptability rules outlined in Table 1, and train a classifier to learn the optimal threshold based on these patterns. For each AC rule, we gather a small set of QA pairs from the NAQT correctness rubrics and error examples from the AC test set and the ROPES *dev* set. We then manually revise QA pairs that violate our AC rubrics to ensure both quality and diversity. For each rule, we create five to ten examples, balancing

the number of correct and incorrect pairs, except for rules that are strictly incorrect.

Seed Examples Each rule is composed of 5 to 10 representative seed examples,where each example is strictly judged based with our rules. However, we do not guarantee that each generated example only corresponds to one rule (may correspond to multiple rules). An example seed example for rule 1 is *question: In 2011, what airport did the most international travelers in North America visit? [SEP] reference: John F. Kennedy International Airport [SEP] candidate: JFK Airport*.

Training data generation We use the manually revised QA pairs as seed examples to prompt GPT-4 with an example and the specific AC rule to generate more similar QA pairs with its judgment of equivalency and correctness.[8] At the end of each generation iteration, we append the generated examples to the seed examples, which we generate a total of 4,234 synthetic AC examples.

Quality validation To verify the automatically generated examples, we manually select fifty generated AC examples and check the judgments. GPT-4's disagreements with humans is most frequent with Rule 2 from Table 1, e.g., GPT-4 often considers numerical answers like *2008-2010 and 2010* to be equivalent. For each generated example, we run GPT-4 self-evaluation by prompting it to verify the correctness of its generated judgments with the rules, questions, reference answers, and candidate answer, and judgment to improved the quality and accuracy of generations (Ren et al., 2023; Zhou and Ai, 2024).

We re-check the 50 generated examples that have inconsistent generated and self-evaluated judgments, and see an improvement of the judgments—39 out of 50 example are consistent with human judgments after self-evaluation. Moreover, we select 300 additional generated examples (mainly examples on Rule 2) and manually check and revise the self-corrected judgments to accommodate our AC rubrics. 180 self-verified judgments are revised and 65 invalid examples (false reference answers, invalid questions) are removed with 3,989 examples remaining in our training material. After two steps of validation, we are confident that the synthetic data mostly align well with our AC rules.

[8]We set the temperature of GPT-4 to 0.9 and frequency penalty to 1.9 to ensure the diversity of generated QA pairs.

Extra Human Annotated Data Augmentation The first step of data augmentation is to collect representative materials to fit the classifier to follow AC norms. We also want to generalize and teach the classifier with more real human annotated judgments. Luckily, we already an existing annotated AC dataset available. We expand the AC learning material by combining our 3,989 synthetic examples with 9,090 examples from the AC training set.

C Model Training Details

We provide specific training details for logistic classifier and fine-tuning details for RoBERTa for ease of reproduction.

Logistic classifier We use `random_state`=*668*, `loss`=*log_loss*, `penalty`=*l2*, `tol`=*1e-3*.

RoBERTa fine-tuning We trained RoBERTa-base with learning rate = *1e-5*, weight decay=*0.01*, batch size=*32* for 2 epochs and save the state with highest accuracy on the AC test dataset following Bulian et al. (2022)'s input training format.

D Evaluation Dataset Details

Table 5 shows examples and number of QA pairs for each of our selected benchmark dataset. NQ-OPEN, HotpotQA, and Narrative-QA are short-form QA, where the reference answers are usually very short, and there are multiple correct reference answers. COQA (Reddy et al., 2019) is a large-scale dataset designed to assess machine comprehension of text through a conversational question and answer format, featuring over 127,000 questions derived from 8,000+ dialogues across seven diverse domains. Each question in CoQA is part of a conversation about a passage, with answers provided in free-form text alongside evidence from the text, challenging systems with tasks like coreference and pragmatic reasoning. BIOMRC (Stavropoulos et al., 2020) is a biomedical machine reading comprehension task dataset that represent real-world machine biomedical reading comprehension tasks. MS MARCO is a question answering dataset of 100,000 real Bing questions, each paired with a human-generated answer. For each of these datasets, we choose a subset by random sampling for evaluation except for COQA, where each conversation contains 10 consecutive questions. We randomly sample 300 conversations from COQA.

E F_1 Score Details

We provide additional formulas as details to calculate the precision, recall, and F_1 score between a reference and a candidate answer. In the context of evaluating a candidate answer against a reference answer, precision, recall, and F_1 score determine the extent of two strings' similarity by tokenizing both answers into individual words. We define tokens as words separated by any white spaces or tabs, where a string *s* can be split to a list of tokens *Tok(s)*. Precision *P* is the ratio of the number of tokens that are present in both the candidate and the reference answer to the total number of tokens in the candidate answer. *P* evaluates the accuracy of the candidate answer by indicating the proportion of its tokens that are relevant to the gold answer:

$$P = \frac{|Tok(candidate) \cap Tok(reference)|}{|Tok(candidate)|} \quad (2)$$

Recall *R* measures the proportion of tokens from the reference answer that are captured by the candidate answer, providing insight into the completeness of the candidate's response:

$$R = \frac{|Tok(candidate) \cap Tok(reference)|}{|Tok(reference)|} \quad (3)$$

F_1 score is the harmonic mean of precision and recall, which balances between the precision and recall metrics. It is particularly useful when the importance of false positives and false negatives are equally significant. The F1 score is computed as follows:

$$F_1 = 2 \times \frac{P \times R}{P + R} \quad (4)$$

The three metrics provide a comprehensive token matching evaluation of the candidate answer's relevance and completeness concerning the gold answer, and are used in many QA model training/evaluation practices mentioned in Table 4.

F *Jeopardy!* Data Collection Pipeline

The *J!Archive* is a fans website to record every season of the *Jeopardy!* game with questions answers presented.[9] The website also includes questions and answers that are answered by *Jeopardy!* players that are originally judged correct/incorrect, but with later revised decisions, where those overruled answers are carefully verified and discussed by audiences or experts. We collected a total of 504 (half correct, half incorrect) such examples from J!Archive use them as the challenge set.

G Comparing Expert answer correctness Judgements

The *Jeopardy!* dataset is a challenge test set that has a very different source of answers with expert judgments and use it to evaluate the effectiveness of various QA evaluation methods on the long-long tail of QA. We introduce this OOD challenge test set that has a very different source of answers with expert judgments and use it evaluate the effectiveness of various QA evaluation methods.

Jeopardy! is a popular American television quiz game show. This show is famous for its distinctive formats of questions, where the questions covers diverse categories, and many of them require a knowledgeable human with years of expertise to be able to answer. An important note is that that there is a pipeline of people who have worked for NAQT have gone on to contribute to *Jeopardy!* in one way or another, the most notable of which is Ken Jennings, who became the first response adjudicator for NAQT (Tournaments, 2019). Thus the selected *Jeopardy!* challenge problems also conform to our AC rules, where we will remove examples with rules we did not adopt (pronunciation, name orders, etc).[10]

Expert dataset format Technically, the responses to *Jeopardy!* clues are "questions" that provide the information sought by the "answers" asked by the host, a gimmick in response to the crucible of the mid-century game show crisis (Stone and Yohn, 1992). Although treating Who is Cleopatra? as a full answer to a question would be a further challenge to AC systems, we treat *Jeopardy!* responses as everything after the verb; in this case, Cleopatra.

Expert dataset details We collect 504 examples from *J!Archive*, a fans website to record every season of the *Jeopardy!* game with questions and answers presented.[11] Each example in the dataset has a question, a reference answer, a candidate response by different *Jeopardy!* players, and expert

[9] https://www.j-archive.com/suggestcorrection.php?clue_id=353154

[10] *Jeopardy!* calls questions clues rather than questions, and the answers are responses. We rephrase the clues to be questions and response to gold answers to conform to machine QA format.

[11] https://www.j-archive.com/suggestcorrection.php?clue_id=353154

judgment of the candidate answer. The dataset includes both difficult correct answers and difficult incorrect answer. E.g., the difficult correct answers are the ones the expert host ruled incorrect but was overruled by a panel. For example, given a question *I've seen the promised land, I may not get there with you, but...we, as a people, will get to the promised land*, the reference answer is Martin Luther King, but Martin Luther King in "I had a dream" was given and ruled correct during the show that later reverted to be incorrect, verified by professionals in accordance with the *Jeopardy!* answer acceptability rules, as the speech in question does not originate from the "I Have a Dream" speech. The dataset includes 50% correct examples and 50% incorrect examples, which we use to test the long-tail automated QA evaluation methods.

G.1 Human Annotations

We delve into details on human annotations in this section. We hired crowdsource annotators from prolific with a 99% approval rate. Each annotator is presented with a set of rules from Table 1. Each rule has an example to justify the judgment shown in Table 6. For the three datasets with short-form answers – NQ-OPEN, NQA, HOTPOTQA, we select 17,602 examples EM judged as incorrect. Specifically, among the 17,272 examples, 6,626 of them are annotated by two different annotators. Among the 6,626 examples with double annotations, 801 (12.1%) annotations disagree (Krippendorff's $\alpha = 0.75$), and we need manually go over those examples to select the better judgment. Thus, the three short-form QA datasets give us 17,272+6,626 = 23,898 examples.

In addition, we select 24,488 examples from the challenge datasets that do not have an exact match with longer reference answers and hired annotators. Thus, our total number of annotations is 23,898+24,488=48,386.

H AC in Other Fields

AC connects many areas of NLP, where many of the following tasks can be considered a variation of answer correctness.

Coreference refers to when two or more expressions in a text refer to the same entity, e.g., person (An and Rudinger, 2023), place, thing (Wu et al., 2020). Equivalence examples are *Her boyfriend* and *Kristy's boyfriend* given question *Who showed up to help Kristy escape when the house was collapsing?*. Since the given question already mentions the name Kristy, *her* in the answer also refers to Kristy's.

Translation involves converting text or speech from one language to another (Bahdanau et al., 2016). Generative models are also likely to provide answers not just in one language. *The capital of France is Paris* and *La capitale de la France est Paris)* are equivalent given question *What is the capital city of France?*. The two answers are equivalent although they are in different languages (English and French). The language of the generated answer is different depending on who is using it.

Logical Reasoning requires the identification of relationships, patterns, or principles underlying the information to answer a question (Nghiem et al., 2024; Weber et al., 2019; An et al., 2023). *reference: he was trying to get back together with Caitlin and candidate: she was informed Dante's planned date with Caitlin* are logically equivalent given the question *Why does Veronica break up with Dante?*. The answers are semantically equivalent under the context of the question. Specifically, the answers require identifying the relationships of three individuals: Veronica, Dante, and Caitlin; it also requires understanding the actions and implications of individuals in the answers. The reference answer suggests Dante has a romantic relationship with Caitlin. The candidate's answer suggests that Veronica has been made aware of Dante's unfaithfulness to her. The determination of the correctness of candidate answers requires the application of logical reasoning between relationships and actions.

I Rule and Question Type Distribution for Individual Dataset

I.1 PEDANTS Simple Insights

We use our trained $F(R)$ and $F(T)$ classifier make rule and question type predictions to the test datasets and show the aggregated rule/question type distributions and PEDANTS' human agreements in Figures 5, 6, 7, and 8 (Appendix I.2). Rule 7 is absent from the test sets, indicating its lack of usefulness. Figure 6 highlights PEDANTS' weakest performance on Rule 2 (numerical information: 75% agreement) and strongest on Rule 1 (entity aliases: 90% agreement). We analyze 30 incorrect Rule 2 examples: over 80% fail due to PEDANTS' inability to recognize the equivalence in date for-

mats, such as when the gold answer is *Feb, 2018* and the candidate answer is *02/2018*, suggesting a critical area for future enhancement in handling numerical and date information. Meanwhile, Figure 8 shows that *when* and *how* question types pose more challenges for PEDANTS. For instance, the *how* question, *How can you increase the battery life of your smartphone?* elicits reference answer *Dim the screen brightness* and candidate *Reduce display luminosity*, which PEDANTS often judges as incorrect, struggling with commonsense knowledge reasoning.

I.2 Rule and Question Type Aggregate Distribution and Human Agreement

Figure 5 and Figure I.2 show the distribution of rules and cfm's human agreement the test datasets. We see an uneven distribution of rules used to evaluate AC. However, the human agreement across rules is quite stable. In addition, we see a lack of examples for rule 5 and rule 7, which suggests that rule 5 and rule style questions are not common in our selected benchmark test sets. Future work can expand test set styles to include more examples for rule 5 and rule 7 to challenge existing automatic QA metrics.

Figure 7 and Figure 8 show the distribution of question types and cfm's human agreement. We see that PEDANTS is quite stable on judging the correctness of different question types, suggesting that the *type* feature is less useful to align PEDANTS with human judgments than the *rule* feature.

I.3 Rule and Question Type Frequency Distribution

Figure 9 shows the rule distribution used to evaluate the correctness of candidate answers for models' answers to individual datasets. Rule 5 (semantic equivalence) and rule 7 (other possible answers) are absent in all data. On the other hand, most of the evaluations fall into rule 3 and rule 4 (more details and fewer details provided), which is frequent since current LLMs tend to generate answers with extra explanations, which often tend to be longer than the reference answers. Figure 10 shows the distribution of question types for different datasets. From the question type distributions, *what* type questions are the most common for all datasets except NQ-OPEN, which has more *who* questions than *what*.

I.4 Rule and Question Type Human Agreement

Figure 11 shows the human agreement with rules used to evaluate the correctness of candidate answers for models' answers to individual datasets. We see that for individual datasets and models, PEDANTS is the worst on Rule 2 (numerical information, dates) and Rule 6 (irrelevant information). PEDANTS still needs significant improvement in judging numerical-type answers. Figure 12 shows human agreement of PEDANTS on different question types for different datasets. From the question type distributions, PEDANTS is quite stable on various question types except for the *why* questions with answers generated by Flan-t5 XL on HOTPOTQA.

J PEDANTS's Weaknesses

We analyzed 45 challenge examples from *Jeopardy!* where all seven methods disagree with experts reveals that commonsense reasoning and fact-checking are the main obstacles for current QA evaluation systems, highlighting the need for more fine-grained AC rules and evaluation data from expert QA community to improve evaluation metrics. Experts evaluate an answer to be correct based on social norms, reasoning, and word choice, which is a much more complicated process than simple string comparison. We list two challenging examples below:

Question: the third largest in our solar system? Reference: Neptune Candidate: Uranus Initially judged incorrect, the *Jeopardy!* panel overruled the decision due to the question's ambiguity regarding the measure of size. *Neptune* has a greater mass than *Uranus*, but *Uranus* has a larger diameter than *Neptune*.

Question: Noted anarchist Prince Peter Alexeivitch Kropotkin wrote a 19th-century entry on this capital? Reference: Moscow Candidate: Saint Petersburg. The answer was also overruled to be correct by the panel with the fact that Kropotkin lived from 1842 to 1921; during his lifetime, the capital of *Russian* was changed from *Saint Petersburg* (1712-1918) to *Moscow* (1918-). Thus, the capital Kropotkin referred to could be either of the answer. Evaluating a hard answer requires validating a fact and reasoning over a fact, which is also a limitation of all current evaluation methods. If we are adopting more data from the Trivia QA community, we should also respect their rules to

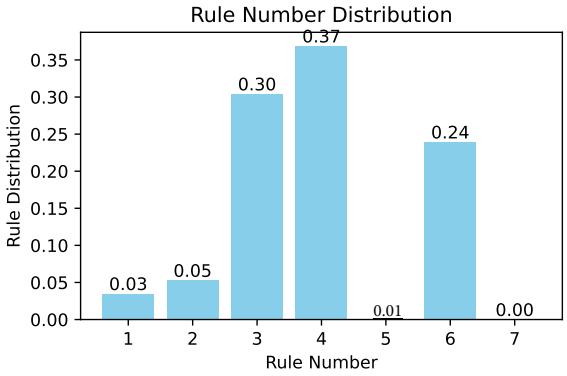

Figure 5: Rule Distribution on all annotated examples. Rule 3 (less details provided) and Rule 4 (more details provided), and Rule 6 (irrelevant information) are the most common rules among our test datasets. There are only under 10 examples for rule 5. Thus, we use 0.01 to signify that there are still some examples for Rule 5.

Figure 6: PEDANTS's human agreement on each rule. We see that PEDANTS is still weak on judging the correctness of dates and irrelevant information, but it is robust at determining entity aliases and when answers have less or more details. We cannot make a proper conclusion for PEDANTS on rule 5 (too few examples).

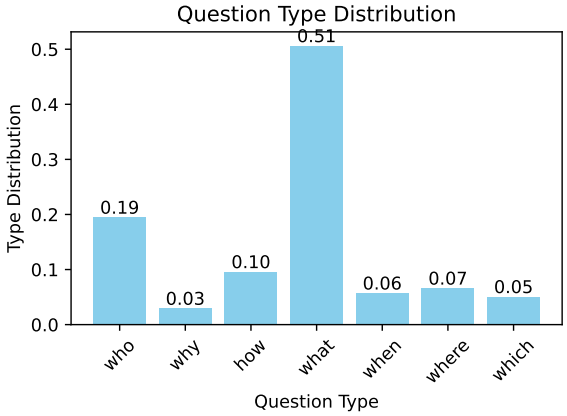

Figure 7: The aggregate question type distribution for all annotated examples. *what* type is the most frequent question type.

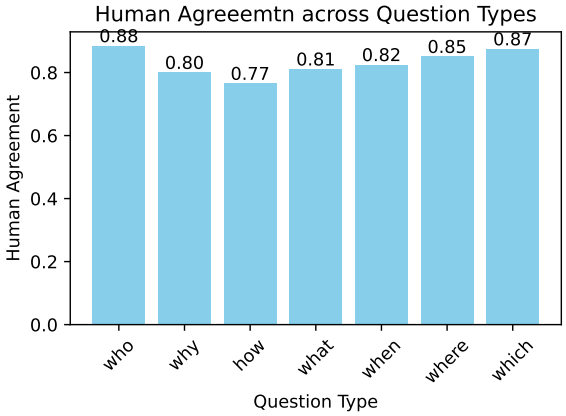

Figure 8: PEDANTS's agreement with humans across question types. PEDANTS is pretty stable with human agreement across various question types.

improve QA evaluation.

K PEDANTS Training Pseudocode

Table 1 shows the pseudocode to train PEDANTS: including rule and question type feature extraction and feature construction for $(q, a, \tilde{a})$ to determine AC.

Algorithm 1 Training and Evaluating PEDANTS

Initialization:
Define question types:
$T = \{\text{who, why, how, what, when, where, which}\}$
Define answer correctness rules $R = \{R_1, R_2, \ldots, R_7\}$
Feature Extraction Classifiers
Train logistic regression $F(T)$ to classify question types T
Train logistic regression $F(R)$ to classify answer correctness rules R
Feature Construction for PEDANTS:
For each QA pair $(q, a, \tilde{a})$:
  Use $F(T)$ to predict a 7x1 probability vector of types T
  Use $F(R)$ to predict a probability vector of rules R
  Calculate token $F1$, precision, and recall for $(a, \tilde{a})$
  Encode $(q, a, \tilde{a})$ using tf-idf
  Concatenate all features into a single feature vector
Collect all feature vectors from each QA pair
Train PEDANTS:
Train logistic regression to determine overall answer correctness using the concatenated feature vectors
Evaluation Stage:
Given $(q, a, \tilde{a})$, PEDANTS predicts either correct or incorrect

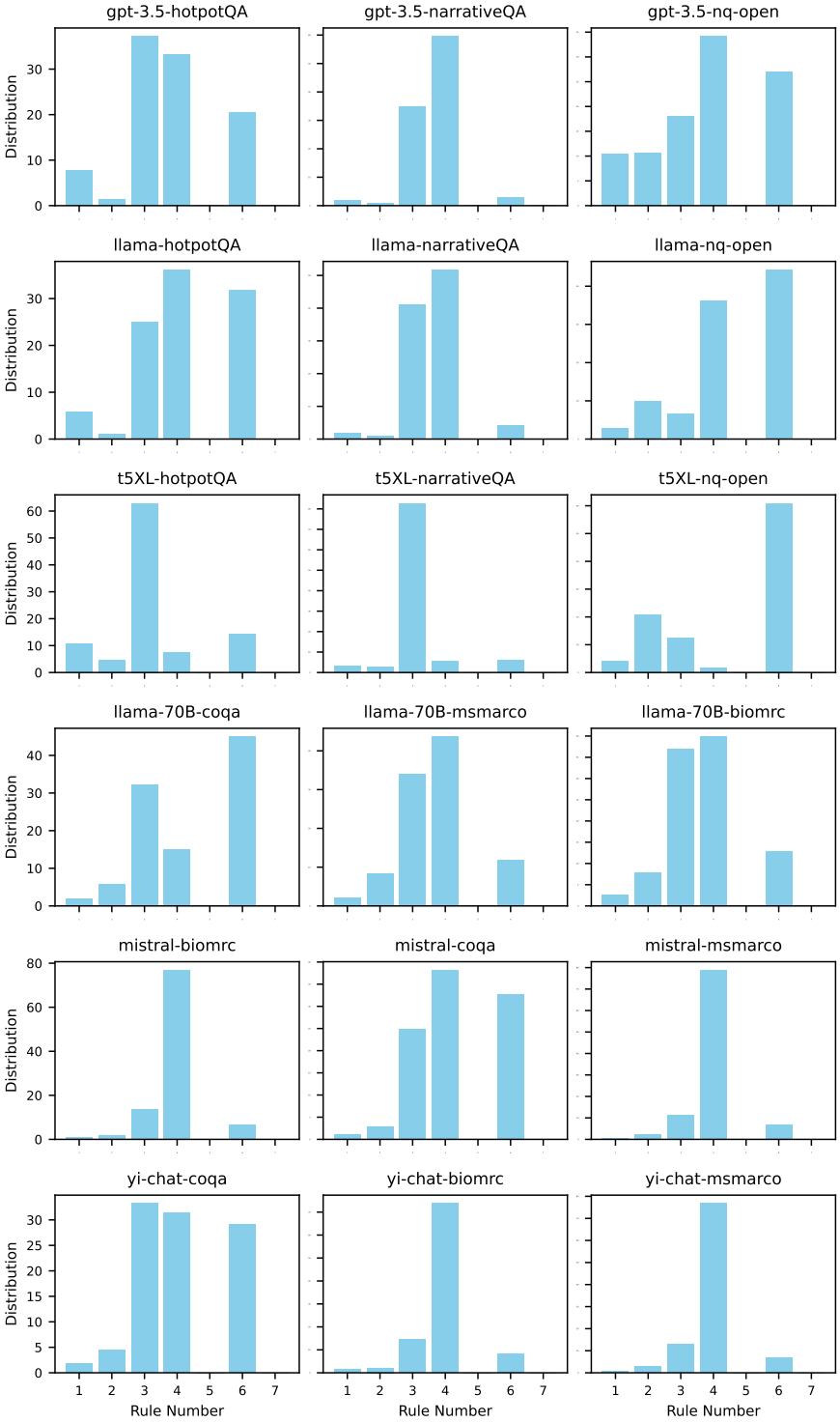

Figure 9: We show the rule distributions used to evaluate for each dataset and each model. Specifically, rule 7 (other possible answers) is absent in any data. On the other hand, most of the evaluations fall into rule 3 and rule 4 (more details and less details provided).

Source	Year	Model	Training Metric	Datasets
Ainslie et al. (2023)	2023	MHA-Large, MHA-XXL, MQA-XXL-GQA-8-XXL	BLEU, F_1	CNN/Daily Mail, arXiv and PubMed, MediaSum, Multi-News, TriviaQA
Jiang et al. (2023)	2023	GPT-3.5, text-davinci-003	EM, F_1	2WikiMultihopQA, StrategyQA, ASQA, ASQA-hint, WikiAsp
Liu et al. (2023)	2023	LongChat-13B, MPT-30B-Instruct, GPT-4.5-Turbo, Claude-1.3, Flan-T5 xxl...	EM	NQ-OPEN
Bevilacqua et al. (2022)	2022	DPR, BM25, GAR, DSI-BART, SEAL (BART-Large)	EM	NQ-OPEN
Zhang et al. (2023)	2023	DPR (FiD and RoBERTa-large), GPT-3.5-turbo	EM	NQ-OPEN, TriviaQA, WebQuestion, HotpotQA
Jeong et al. (2024)	2024	Flan-T5 xl (3B), Flan-T5 xxl (11B), GPT-3.5-Turbo	EM, F_1	NQ-OPEN, TriviaQA, SQuAD
Chen and Tanrikulu (2024)	2024	Electra Small Discriminator	EM, F_1	Adversarial SQuAD, TriviaQA, SQuAD
Zhu et al. (2023)	2024	BERT, GPT-3.5, T5-B, T5-L, TB-T5-L	F_1	HotpotQA, DROP (Dua et al., 2019)
Zhu et al. (2023)	2022	RoBERTA, BART, ELECTRA (Clark et al., 2020), T5	EM, F_1	SQuAD, NewsQA, TriviaQA, SearchQA, HotpotQA, NQ-OPEN, DROP, RACE...
Luo et al. (2023)	2023	LLaMA (7B), Vicuna (13B), ChatGPT	GPT-4-Eval	CommonsenseQA (Talmor et al., 2019), OpenbookQA (Mihaylov et al., 2018), ARC-Challenge (Clark et al., 2018)
Liu et al. (2024)	2024	GPT-3.5, GPT-4, Llama3-ChatQA	EM, F_1	COQA (Reddy et al., 2019), ChatRAG BENCH (Liu et al., 2024)
Xu et al. (2024)	2024	Gemma, LLama3 GPT-4, Phi-3	EM, F_1	(Xu et al., 2024)
Yona et al. (2024)	2024	PaLM 2	BLEURT	GRANOLA (Yona et al., 2024)
Meta Platforms, Inc. (2024)	2024	LLaMA-3.1-8B	EM, F_1, accuracy, pass@1 (coding)	Llama-3.1-8B-evals

Table 4: Question answering model training papers and the evaluation metrics used during training. In the year of 2024, most of the QA training papers still use EM and F_1 as the evaluation metrics.

Dataset	# Pairs	Context	Question	Gold Answer
NQ-OPEN	3,610	-	who won the American league east in 2017	The Yankees, Houston Astros
HotpotQA	3,300	Asmara International Airport (IATA: ASM) (ICAO: HHAS). Asmara currently hosts the country's only operating international airport...	Asmara international airport is in which country?	Eritrea
Narrative-QA	3,300	The narrator, a Bostonian, returns after a brief visit a few summers prior, to the small coastal town ... cMaine coastal town increases each day.	What was Littlepage's job?	sailor, retired sailor
COQA	5,500	CHAPTER XXX. THE MEETING FOR RAIN. Meanwhile the Auld Lichts were in church, waiting for their minister...	Who gaped at Hendry?	Peter was so taken aback that he merely gaped at Hendry
BIOMRC	1,373	Why is a second dose of MMR necessary? About $2\% - 5\%$ of persons do not develop measles immunity after the first dose of vaccine. This occurs for a variety of reasons. The second dose is to provide...	Why is a second dose of mmr necessary	To provide another chance to develop measles immunity for persons who did not respond to the first dose.
MS MARCO	3,000	It is located on a 100-acre (0.40 km 2) site on the northern quay of the Royal Victoria Dock in London Docklands, between Canary Wharf and London City Airport. Phase II was...	where is excel arena london	It is located on a 100-acre (0.40 km 2) site on the northern quay of the Royal Victoria Dock in London Docklands, between Canary Wharf and London City Airport.
MOCHA	5,033	Somewhere in me I knew it all along, there are all those moments when he stares into my eyes and his start to sparkle while this gorgeous grin spreads across his face...	What's a possible reason the guy stares into the writer's eyes ?	Because he likes her a lot/He's a child and it's a very rare thing.

Table 5: The seven datasets we use for our generalization testing. The later three datasets generally have much longer gold answers than the previous three, which are considered more challenging. We do not include contexts for NQ-OPEN for more variability of generated answers to challenge the evaluation methods. MOCHA already includes human annotated labels.

Prompt template for generating answer correctness examples
You will be provided with a specific rule to assess the correctness of a candidate answer given a reference answer. Your assignment involves generating high quality diverse example questions different from the given example. Each example must include a question, a reference answer, a candidate answer, and a judgment on whether the candidate answer is correct based on the provided rule. It is important that your examples follow the structure of the given example, with an emphasis on ensuring that the reference and candidate answers are similar but not identical. **[Example]** Rule: Selected Rule Your Seed Example **[Instruction]** 1. Try to include hard negative and hard positive examples in your responses. 2. Try to make your responses as diverse as possible. 3. Only generate json format responses. Rule: Selected Rule **[Your Response]**

Table 6: The prompt template for generating answer correctness examples. The red texts are your inputs.

Prompt template for self-verifying

You are given a correctness rule, a question, a reference answer, and a candidate answer to the question. Your task is to determine the correctness of the candidate answer by outputting either **correct** or **incorrect**.

Rule: Corresponding Rule

[Example]
Question: Example question
Reference: Example reference answer
Candidate: Example candidate answer
Judgment: Gold judgment

[Input]
Question: Example question
Reference: Example reference answer
Candidate: Example candidate answer
Judgment:

Table 7: The prompt template for self-verifying the generated examples.

Prompt template for GPT-4-Eval

[1] You will be given a set of rules and representative examples for each rule to determine whether the candidate is correct based on the question and the reference answer.
[2] Then you will be given a question, a reference answer, and a candidate answer. You tasks is to determine whether the candidate is correct based on the question, the reference answer, and the rules and examples. Please only output the rule number and **"correct"** or **"incorrect"** as the output.

Rule 1: Widely recognized aliases, pseudonyms, or alternative names that are commonly associated with the reference answer entities are acceptable.
Example 1: Question: Who plays Red on Orange is the New Black? reference Answer: Kate Mulgrew. candidate Answer: Katherine Maria Mulgrew. Your Answer: Rule 1,correct.
Rule 2: Exact dates, years, numerical are required unless the question specifically asks for approximations.
Example 2: Question: How tall can a giraffe grow? reference: 16-20 feet. candidate: 18 feet. Your Answer: Rule2,incorrect.
Rule 3: The candidate answer provides less details but should include the essential and correct information required by the question (specificity 1).
Example 3: Question: What followed the late local programming after Super Bowl 50? reference: The Late Late Show with James Corden. candidate: The Late Show. Your Answer: Rule 3,incorrect.
Rule 4: candidate answer contains the reference answer or an equivalent part. The additional information must be factually correct and does not contradict the question or reference answer (Specificity 2).
Example 4: Question: When did Morales launch his policy in the eastern lowlands? reference: 2009. candidate: August 3, 2009. Your Answer: Rule4,correct.
Rule 5: A high degree of word overlap or similarity does not establish equivalence. The answer must be contextually and semantically accurate.
Example 5: Question: What is the primary source of energy for the Earth? reference: Solar radiation from sun. candidate: Solar energy from sun. Your Answer: Rule 5,incorrect.
Rule 6: The candidate answer is irrelevant to question and the given answer or is an inaccurate description of the question should be incorrect.
Example 6: Question: Which event occurred first, the first Super Bowl or the formation of the Premier League? reference: first Super Bowl. candidate: formation of the Premier League. Your Answer: Rule 6,incorrect.
Rule 7: candidate response is correct but not in the reference list.
Example 7: Question: When did the Golden State Warriors win the NBA Championship? reference: 2015, 2017. candidate: 2016. Your Answer: Rule 7,correct.
Please output your answer below.
Question: **[Input question]**
reference: **[Input reference answer]**
candidate: **[Input candidate answer]**
Your Answer:

Table 8: The prompt template for GPT-4-Eval. We include representative examples for each rule to align GPT-4's judgment with the rules.

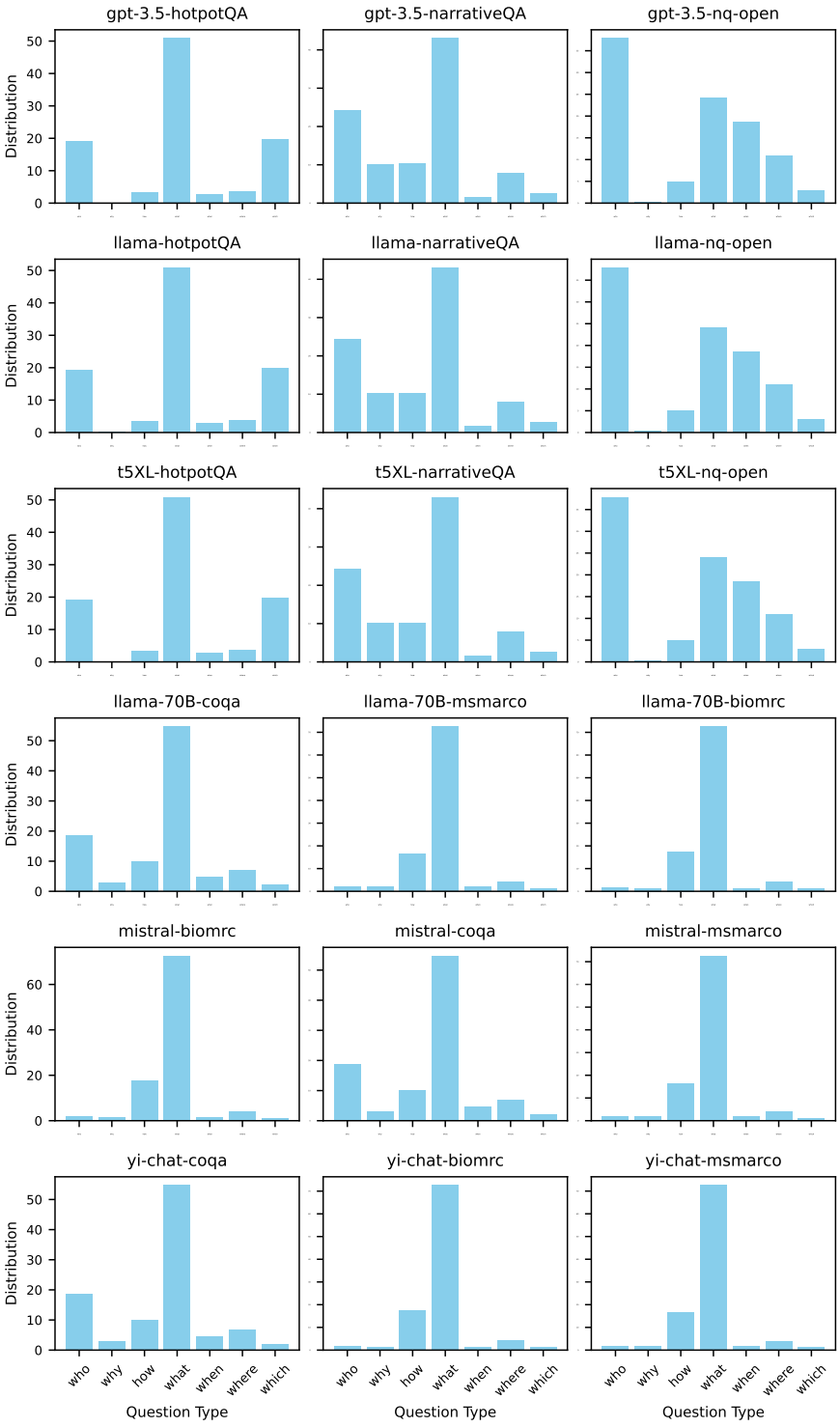

Figure 10: We show the question type distributions used to evaluate each dataset and each model. Specifically, *what* type questions are the most frequent across all datasets.

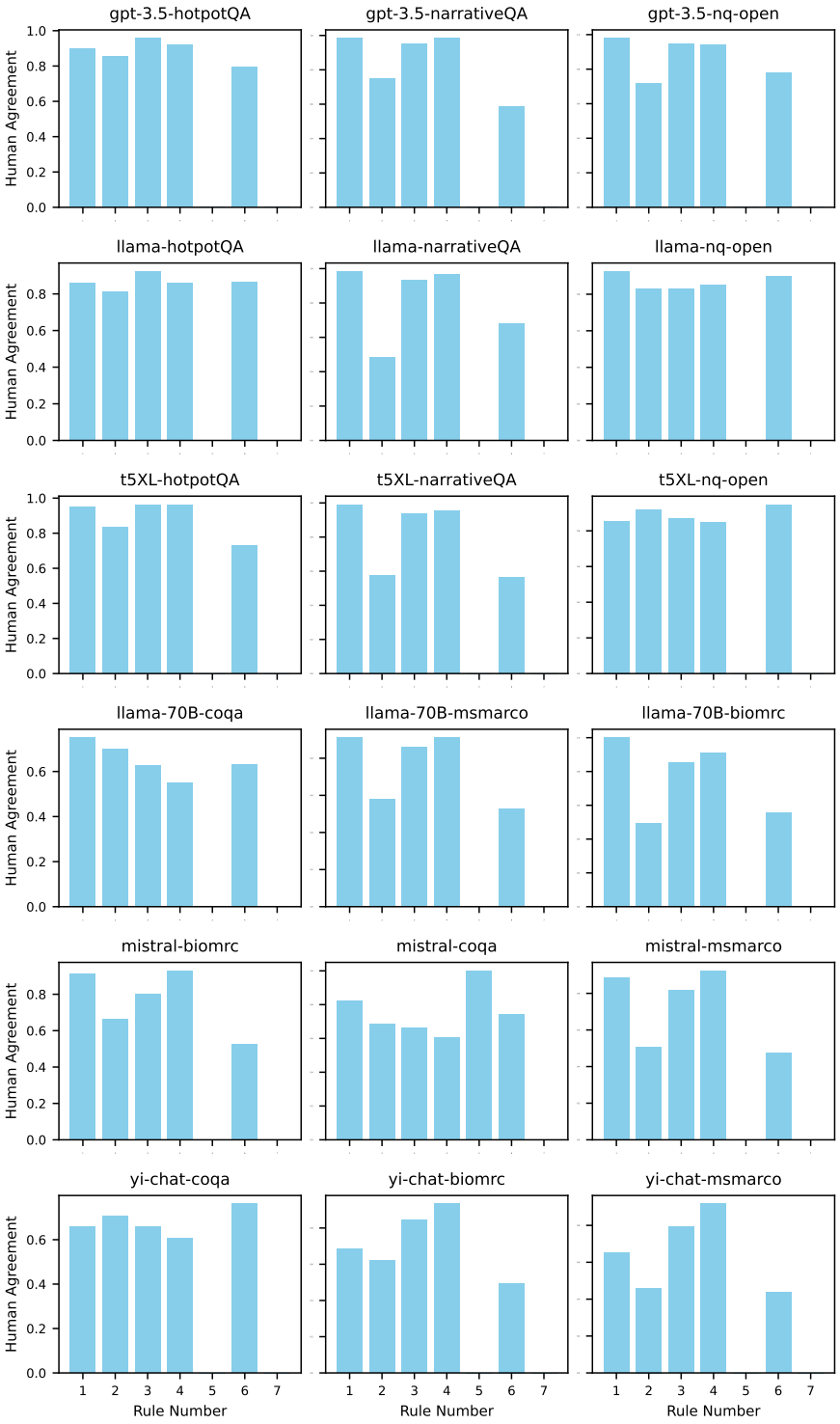

Figure 11: We show PEDANTS's human agreement for different rules across individual datasets and models. PEDANTS has its limitation in judging Rule 2 (numerical information) and Rule 6 (irrelevant information), suggesting a future improvement of adding more meaningful training data for rule 2 and rule 6.

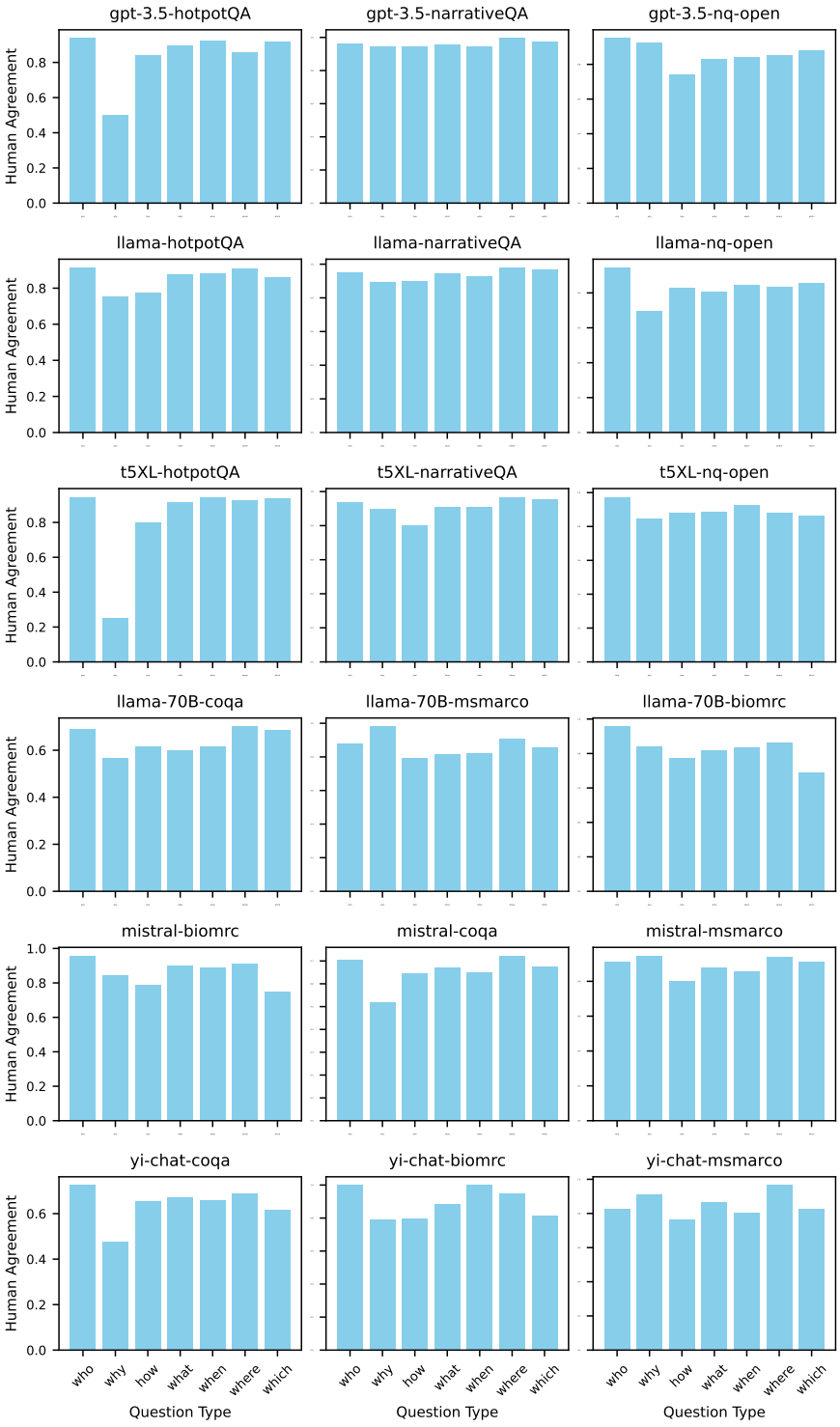

Figure 12: We show PEDANTS's human agreement for question types across individual datasets and models. PEDANTS is stable on various question types.