QUALEVAL: Qualitative Evaluation for Model Improvement

Anonymous submission

Abstract

Quantitative evaluation metrics have been pivotal in gauging the advancements of AI systems like large language models (LLMs). However, due to the intricate nature of real-world tasks, a single scalar to *quantify* and *compare* performance trivializes the fine-grained nuances of model behavior. Additionally, metrics do not yield actionable diagnostics for model improvement, thus requiring extensive manual efforts of scientists, involving sifting through vast datasets and attempting hit-or-miss adjustments to training data or setups. In this work, we address the shortcomings of quantitative metrics by proposing QUALEVAL, which uses automated *qualitative* evaluation as a vehicle for model improvement. QUALEVAL uses a powerful LLM reasoner and our novel flexible linear programming solver to generate human-readable insights that when applied, accelerate model improvement. The insights are supported by a dashboard report with fine-grained visualizations and human-interpretable analyses. We corroborate the faithfulness of QUALEVAL by demonstrating that leveraging its insights, for example, improves the absolute performance of the Llama 2 model by up to 15% points relative on a challenging dialogue task (DialogSum) when compared to baselines. QUALEVAL successfully increases the pace and quality of model development by eliminating the need of arduous manual analysis, thus serving as a data-scientist-in-a-box.

1 Introduction

The recent success of large language models (LLMs) while can be attributed to data and compute scaling, has also been the result of evaluation metrics that allow benchmarking and comparison of models. This surge in the development of LLMs and associated tasks has reignited the need for innovative evaluation methods, aiming to provide more effective guidance throughout the model development process (Tornede et al., 2023; Paranjape et al., 2022). Traditional scalar quantitative metrics like perplexity, BLEU, and ROUGE play an important role in objectively measuring improvements in model performance. However, these scalar metrics cannot capture the nuances of model behavior and therefore are unable to provide model developers actionable directions and diagnostics for model improvement (Novikova et al., 2017; Liu et al., 2016). In practice, this deficiency necessitates model developers to collaborate with an army of data scientists and engineers to iterate over a diverse array of models and tasks, especially in rapidly evolving real-world settings.

In this work, we use "quality over quantity" as a guiding principle to propose our model and task agnostic method QUALEVAL, that uses qualitative evaluation to address the issues with quantitative metrics. Given a model that is being developed for a task, QUALEVAL serves as an automated data scientist by analyzing the dataset and the model's predictions to generate actionable directions that improve the model, supported by a comprehensive dashboard containing fine-grained analysis of the model's behavior (Figure 1). The directions identified by QUALEVAL to improve the model significantly expedite the model development lifecycle. Rather than rejecting the use of metrics, QUALEVAL uses them as just one of the parts of a more holistic and useful evaluation.

QUALEVAL's algorithm for facilitating model improvement can be broken down into three steps (Figure 2): (1) *Attribute discovery*: Automatic discovery of domains and sub-tasks in the dataset, to help identify issues at a fine-grained level. (2) *Attribute assignment*: Utilize a novel flexible linear programming solver to assign attributes to instances and analyze the performance on different attributes to create a human-readable dashboard. (3) *Insight generation*: Parse the generated dashboard to provide natural language insights that improve the model. QUALEVAL's end-to-end pipeline is completely automated and requires no human intervention.

We demonstrate QUALEVAL's potency on a wide range of tasks including code generation, dialogue summarization, and multiple-choice question answering. We harness these insights provided by QUALEVAL to *precisely* and significantly improve the performance of the open-source Llama 2 model on a dialog summarization task. In a demonstration of efficacy, QUALEVAL's insights allow a model practitioner to make changes to the fine-tuning procedure by augmenting with the *right* instances, thus leading to an over-

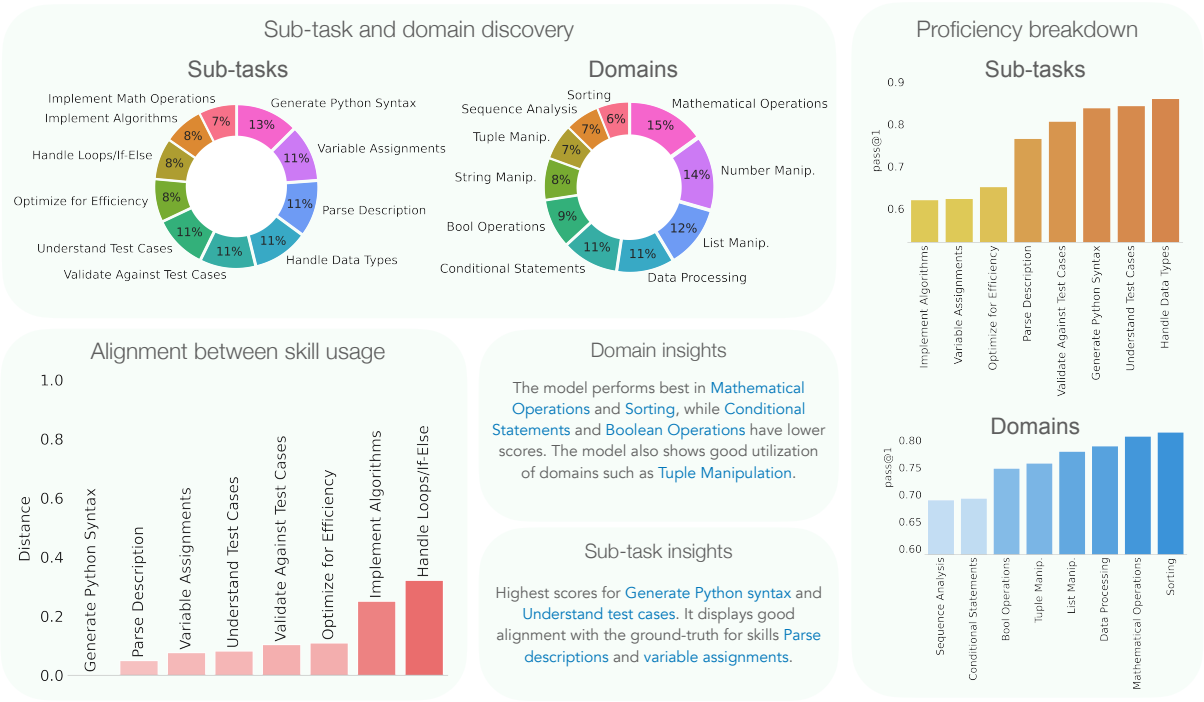


Figure 1: QUALEVAL goes beyond a single scalar metric and provides a dashboard that helps understand the model's performance in a fine-grained manner. QUALEVAL's insights are faithful and lead to accelerated performance improvement when applied to the model. The dashboard visualizes the performance of the DAVINCI-3 model on MBPP.

all ROUGE-L score improvement of 15%. QUALEVAL's insights also allow targeted improvement in sub-domains of critical importance, with absolute improvements of up to 5 points.

Our contributions are as follows: (1) We propose the first qualitative evaluation framework for LLMs. (2) We introduce a novel and faithful flexible linear programming-based algorithm to automatically and accurately assign attributes to input instances, which are consequently used to generate faithful and actionable insights. (3) We demonstrate that the generated insights can be effectively leveraged for model improvement, leading to accelerated model development.

2 Methodology

2.1 Formulation

Quantitative evaluation Quantitative evaluation is the standard approach to evaluating models based on averaging the value obtained by using a metric to evaluate instances of the dataset independently. Formally, given a dataset $\mathcal{D}$ comprising of instances containing inputs (x_i) and ground truth outputs (y_i), a performance (proficiency) metric $\mathcal{M}$, and a model f, then:

$$\mathcal{D} \coloneqq \{(x_i, y_i)\}_{i=1}^{N}$$
$$\hat{y}_i \coloneqq f(x_i)$$
$$\mathcal{M} \colon (x_i, y_i, \hat{y}_i) \to \mathbb{R}$$
$$\text{Quantitative evaluation} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathcal{M}(x_i, y_i, \hat{y}_i)$$

Qualitative evaluation Qualitative evaluation (QUALEVAL) is based on holistically evaluating the model's performance in a fine-grained manner rather than relying on a single scalar value. QUALEVAL outputs a detailed dashboard that describes the intricate nature of the model's performance with the direct goal of improving it by providing actionable insights. QUALEVAL backs up the insights with relevant evidence like visualizations and human-readable reasoning. Formally, let $\mathcal{V}$ be the vocabulary of the language of the dashboard (here, English) and $\mathcal{I}$ be the set of all possible visualizations. Let $\mathcal{E}$ be an evaluator system which generates the dashboard, where the system includes LLMs to reason and provide insights and image generation models to generate plots for example. Then, the output of QUALEVAL is given by:

$$\mathcal{E} \colon \{x_i, y_i, \hat{y}_i\}_{i=1}^{N} \to (\mathcal{V} \cup \mathcal{I})^{\star}$$
$$\text{Qualitative evaluation} = \mathcal{E}\left(\{x_i, y_i, \hat{y}_i\}_{i=1}^{N}\right)$$

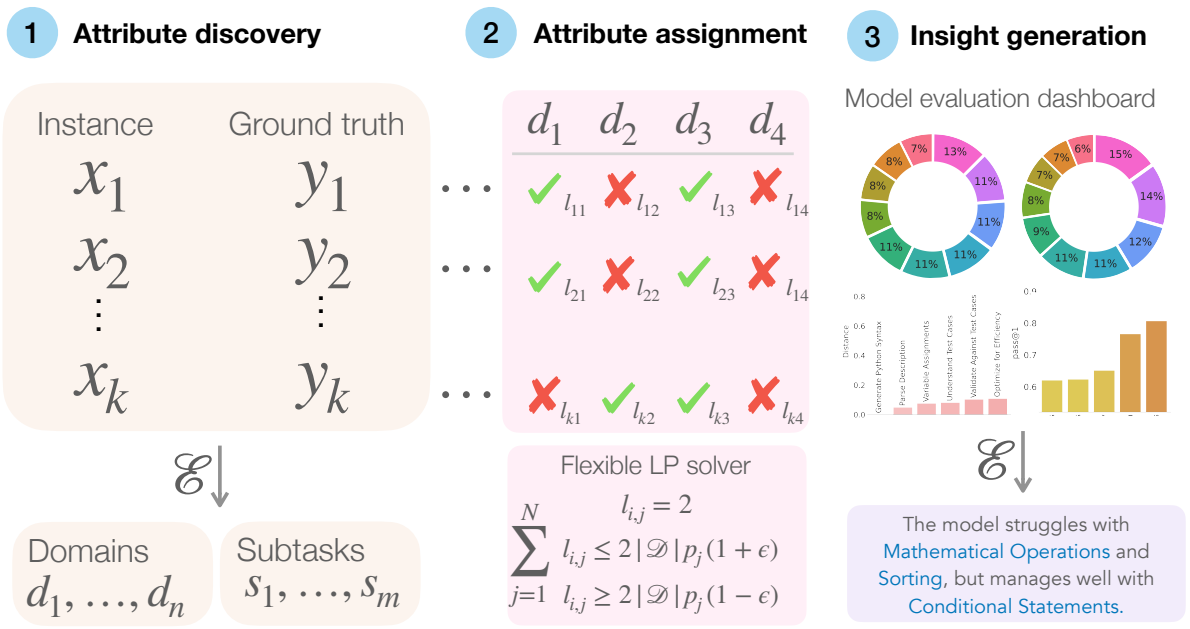


Figure 2: QUALEVAL automatically discovers domains and sub-tasks from input data through an evaluator LLM, $\mathcal{E}$. QUALEVAL then automatically assigns 2 domains and 2 sub-tasks to every sample in the dataset by solving a flexible linear program. Finally, QUALEVAL generates a comprehensive dashboard and presents interpretable and actionable insights for practitioners.

Qualitative evaluation does not reject the use of metrics but uses them as one of the parts of a more actionable evaluation. In essence, quantitative evaluation is just a small subset of QUALEVAL.

2.2 QUALEVAL: Qualitative evaluation

QUALEVAL consists of multiple steps that help provide interpretable and actionable insights and we break them down below.

Attribute discovery Given the dataset $\mathcal{D}$, QUALEVAL uses an evaluator LLM ($\mathcal{E}$) to automatically discover relevant domains and sub-tasks, $d_1 \cdots d_N$ and $t_1 \cdots t_N$ in the dataset. We refer to these domains and sub-tasks as attributes. Specifically, we prompt $\mathcal{E}$ with the dataset and a task instruction signifying how to solve the dataset ($Instr_D$) to generate the attributes (see A.3 for the exact prompt). Given that datasets can have a large number of instances and LLMs have context length limits, we iteratively sample k instances from the dataset and repeat the prompting process $\frac{|\mathcal{D}|}{k}$ times to generate a large list of attributes ($d_1 \cdots d_M$, $t_1 \cdots t_L$). To ensure that we choose high-quality attributes, we prune the list of candidates in an iterative process by reducing the size by a factor of $p > 1$ in each turn and repeating the process until we have N attributes. In each step, we prompt $\mathcal{E}$ to shrink the list by choosing the best attributes from the previous list of candidates. Therefore, this iterative scalable procedure allows QUALEVAL to discover attributes in arbitrarily large data across a wide range of tasks, notwithstanding the context window limitations of $\mathcal{E}$.

Attribute assignment QUALEVAL performs attribute assignment ($d_1 \cdots d_N$ and $t_1 \cdots t_N$) by scoring the "affinity" or relevance of each instance with different attributes. Let $s_{i,j}^{domain}$ and $s_{i,j}^{task}$ denote the domain and sub-task affinity scores, where $i \in \{1 \cdots |\mathcal{D}|\}$ and j denotes the number of attributes ($\{1 \cdots N\}$).

We use a novel flexible linear programming solver to perform the attribute assignment by ensuring the following properties: (1) An instance is assigned 2 domains and sub-tasks each so that we can give concrete insights. (2) The number of assignments to an attribute is proportional to the prior probability of the attribute. This ensures that rare attributes are not ignored. (3) Choose the assignments with maximum affinity for each instance. We achieve the above wish-list by formulating the attribute assignment as a linear programming (LP) problem.

Given the affinity scores and the prior probabilities, p_i, we assign every sample to 2 domains and 2 sub-tasks. However, we want the assignments to respect the prior probabilities i.e. ratio of the number of assignments to all the attributes should be equal to the ratio between the prior probabilities. We enforce this by constraining the number of assignments to an attribute to be $p_i \times |\mathcal{D}| \times 2$.

Let $\mathbf{l}$ be the assignment matrix, where $l_{i,j} = 1$ indicates that the i^{th} sample is assigned to the j^{th} attribute and $l_{i,j} = 0$ indicates otherwise. Let p_j be the prior probability of the j^{th} attribute. To accommodate for the noisiness in an automated method, we make the prior probability constraint flexible by adding some slack, $\epsilon \times p_j \times |\mathcal{D}| \times 2$ ($\epsilon = 0.1$) so that QUALEVAL has some wriggle room to change the attribute probability dis-

tribution in favor of better assignments. Therefore, to enforce the prior probability constraint, we sum across the columns of $\mathbf{l}$ and constrain the sum to be between $2 \times |\mathcal{D}| \times p_j \times (1-\epsilon)$ and $2 \times |\mathcal{D}| \times p_j \times (1+\epsilon)$. To ensure we assign each sample to 2 attributes, we sum across the rows of $\mathbf{l}$ and constrain the sum to be 2. We formalize the LP as:

$$\max_{\mathbf{l}} \sum_{i=1}^{N} \sum_{j=1}^{N} l_{i,j} s_{i,j}^{domain/task}$$

$$\sum_{j=1}^{N} l_{i,j} = 2 \quad \forall i \in \{1 \cdots |\mathcal{D}|\}$$

$$\sum_{i=1}^{N} l_{i,j} \leq 2 * |\mathcal{D}| * p_j * (1+\epsilon) \quad \forall j \in \{1 \cdots N\}$$

$$\sum_{i=1}^{N} l_{i,j} \geq 2 * |\mathcal{D}| * p_j * (1-\epsilon) \quad \forall j \in \{1 \cdots N\}$$

$$l_{i,j} \in \{0,1\} \quad \epsilon < 1 \quad \forall i,j \in \{1 \cdots N\}$$

We perform an expert verification of the attribute assignments by sampling 100 samples from the dataset and asking three machine learning practitioners if both the domain and sub-task assignments are correct and find that they are indeed correct on average 84% and 90% of the time.

Once we have the assignments, we evaluate each instance using the proficiency metric $\mathcal{M}$ for each domain and sub-task to get $\mathcal{M}(x_i, y_i, \hat{y}_i)$. We use the assignments to breakdown the proficiency metric by domains and sub-tasks and automatically generate visualizations that help understand the model's fine-grained performance.

Measuring sub-task skill alignment For several datasets, predicting the right answer is not good enough, and producing an answer that uses the same sub-tasks as the ground truth is important. We call this skill alignment and compute it by measuring the correlation between the sub-task affinity scores of the ground truth and the model prediction (higher values implying higher skill alignment).

Insight generation QUALEVAL then leverages the visualizations from previous stages to generate useful and actionable insights as a natural language output. We prompt $\mathcal{E}$ with the data from the prior probability, proficiency breakdown, and skill alignment visualizations to generate useful insights (See A.3 for exact prompt). We integrate all the visualizations and insights into a human-readable dashboard depicted in Figure 1.

3 Experimental Setup

Datasets We evalaute QUALEVAL on three datasets: MBPP (Austin et al., 2021) (sanitized), DialogSum (Chen et al., 2021), and MMLU (Hendrycks et al., 2020) (clinical knowledge split). MBPP and DialogSum involve generative tasks and involve generating a Python program from a prompt and summarizing a conversation respectively. MMLU contains a wide range of multiple-choice questions from different domains and we pick the clinical knowledge split to evaluate our model on knowledge-intensive tasks. We use the same evaluation splits as the original papers and use the test splits for MBPP and MMLU and the validation split for DialogSum. We use the pass@1, ROUGE-L, and accuracy as proficiency metrics for MBPP, DialogSum, and MMLU respectively.

Models We use both closed and open-sourced models: CURIE, DAVINCI-2, and DAVINCI-3 models from OpenAI and Llama 2 (7 billion chat models (Touvron et al., 2023)). We use a temperature of 0.9 for all models and use two randomly sampled in-context samples for prompting models unless mentioned otherwise. We instantiate $\mathcal{E}$ with the GPT-3.5-TURBO model (OpenAI, 2023).

Llama fine-tuning We use LoRA (Hu et al., 2021) to efficiently fine-tune the Llama 2 7 billion parameter model and train with 8 bit precision. We sweep over five learning rates ($2e-5$, $5e-5$, $1e-4$, $2e-4$, $1e-3$) and pick the checkpoint with the best validation performance. We train for up to 400 steps and we use a constant learning rate schedule.

Attribute generation We set N (the initial number of generated categories) to 15, p (the pruning factor) to 4, and k (the number of few-shot examples during category generation) to 5 in our experiments.

4 Results

We systematically present different aspects of our dashboard. Firstly, we show that attribute discovery (domains and sub-tasks) of QUALEVAL is well-grounded and faithful to the dataset. Secondly, we show that QUALEVAL's flexible LP solver correctly assigns attributes to instances of the dataset, allowing it to perform meta-reasoning over different domains and sub-tasks. Finally, we validate that the concise natural language insight generated leads to improvement in the model's performance.

4.1 Discovering Domains and Sub-tasks

Discovering the latent domains and sub-tasks in a dataset and understanding their prominence through the prior probability of their occurrence is a critical step for QUALEVAL. QUALEVAL performs both the discovery and prior-probability computation *automatically* and *faithfully*.

As an example, Figure 3 presents the prior probabilities of the domains and sub-tasks in the MBPP and DialogSum datasets. We find that the MBPP dataset comprises a large set of samples that involve domains like mathematical/numerical operations (29%) and list manipulation (12%) while domains like sorting (6%) and tuple manipulation (7%) are less prevalent. Interestingly, QUALEVAL captures fine-grained nuances

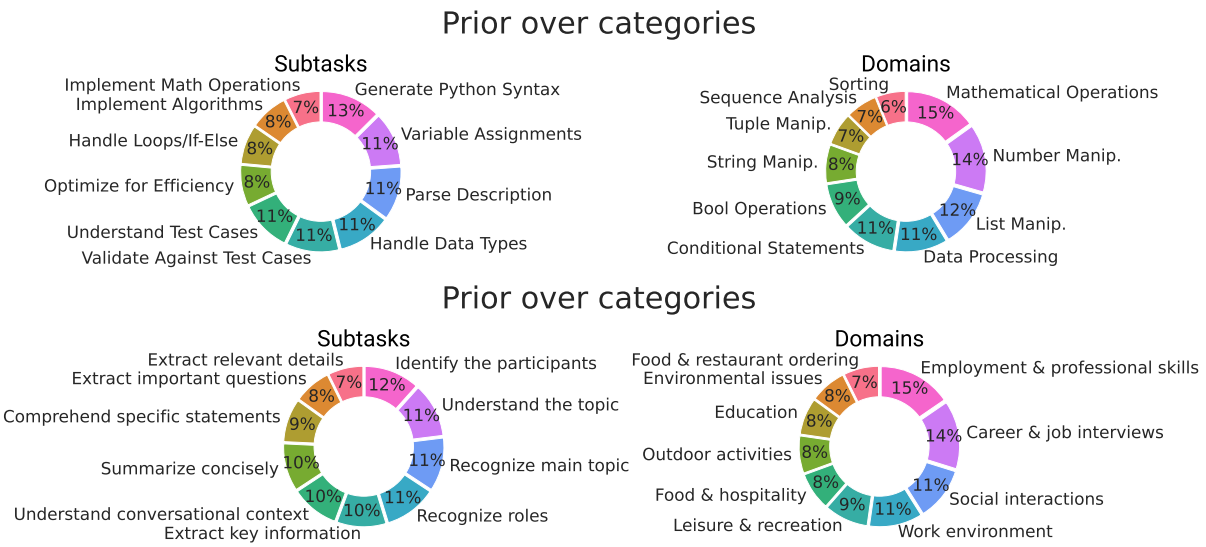


Figure 3: Prior probabilities of domains and sub-tasks on the MBPP (top) and DialogSum (bottom) datasets

by including closely related yet different sub-tasks like "Implement mathematical operations" and "Implement algorithmic operations", giving practitioners a nuanced understanding of their evaluation data.

As another illustration (Figure 3 bottom), the DialogSum dataset is dominated by samples involving domains like employment and professional skills (15%) and career and job interviews (14%), while domains like education and outdoor activities are less prevalent (8% and 8% respectively). Though the overall food domain is also frequent, it is listed under two fine-grained domains, "Food and restaurant ordering" (7%) and "Food and hospitality" (8%), which further highlights QUALEVAL's ability to capture fine-grained nuances. The evaluation also suggests the dominance of sub-tasks that involve identifying the participants (12%), understanding and recognizing the main topic (22%), and recognizing the roles in the conversation (11%), which are conceptually important sub-tasks for summarizing a conversation between two people.

Faithfulness of priors We show that the attributes discovered and prior probabilities assigned are faithful to the dataset. While most datasets do not have ground truth annotations for the domains and sub-tasks, (Pal et al., 2022) introduces a multiple-choice question answering dataset, MedMCQA, collected from real-world medical exam questions, and includes domain annotations. We randomly sample 250 questions from the MedMCQA dataset and leverage QUALEVAL to discover domains and find the prior probabilities. We compare the prior probabilities from QUALEVAL with the ground truth domain annotations from MedMCQA in Figure 4. We find that the domain priors from QUALEVAL are highly aligned with the ground truth annotations ("Pediatrics" (9% vs 9%), "Obstetrics and Gynecology"(6% vs 7%), and "Pharmacology" (6% vs 6%) and "Microbiology" (4% vs 6%)). Interestingly, QUALEVAL splits the "Dental" domain into more precise domains such "Dentistry", "Dental Hygiene", "Dental procedures", and "Dental anatomy", further highlighting QUALEVAL's ability to capture hierarchies and nuances in the data.

4.2 Proficiency categorized by Domains and sub-tasks

To generate useful insights, one needs a clear understanding of the model's proficiency in the various domains and sub-tasks, and we demonstrate that QUALEVAL provides exactly this. QUALEVAL leverages the domain and sub-tasks assignments generated from our flexible LP solver to get a precise breakdown of the proficiency of a model for different domains and sub-tasks.

Figure 5 highlights the proficiency of the DAVINCI-3 model on domains like sorting, mathematical operations, and data processing and on sub-tasks like handling data types, understanding test cases, and generating Python syntax. We find that QUALEVAL's categorization and proficiency judgement is faithful and aligned, as corroborated by analysis from Austin et al. (2021) that also suggests that models on MBPP perform well on "coding interview" type questions which generally involve data structures, sorting, list manipulation, and data processing.

Austin et al. (2021) also suggests that models struggle with samples related to advanced math problems and samples with multiple sub-problems. This conforms with QUALEVAL's proficiency breakdown which reveals that the model struggles with samples involving the "Implement algorithms" and "Variable assignments" sub-tasks and the "Conditional statements" and "Sequence Analysis" domains, which are often leveraged to solve math problems and samples with multiple sub-problems. These findings serve to reinforce the distinctive capability of QUALEVAL in of-

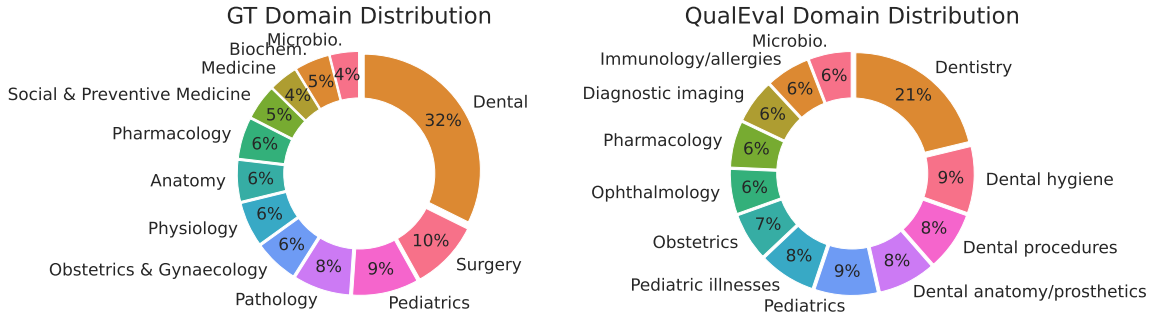


Figure 4: QUALEVAL faithfully discovers and scores attributes. We compare the domain priors discovered by QUALEVAL(right) with the ground truth domain annotations (left) in the MedMCQA dataset and find a high degree of alignment (e.g., "Pediatrics" – 9% vs 9%, "Obstetrics and Gynecology" – 6% vs 7%, and "Pharmacology" – 6% vs 6%).

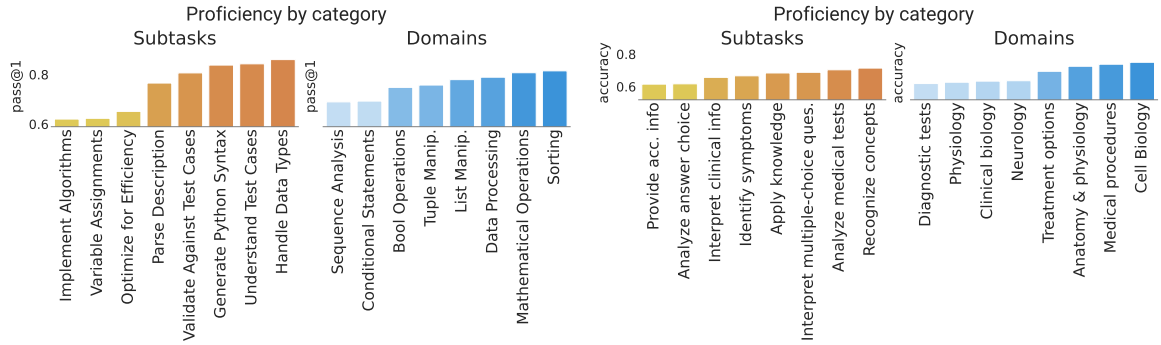


Figure 5: Proficiency breakdown for different sub-tasks and domains in the MBPP and MMLU (clinical knowledge) datasets for DAVINCI-3.

fering a precise and nuanced comprehension of model proficiency, made possible by our flexible LP solver.

QUALEVAL is task-agnostic, with our flexible LP solver making it potent even in niche domains such as clinical data. Figure 5 demonstrates high proficiency of the DAVINCI-3 model on the cell biology and medical procedures domains and sub-tasks related to analyzing and processing medical test data and recognizing key terms/concepts in clinical biology. However, the model struggles with sub-tasks related to providing accurate information and analyzing the correct answer choice.

4.3 Interpretable and Actionable natural language insights

To aid model developers in understanding the dense fine-grained analysis in the prior sections, we present interpretable and actionable natural language insights grounded in the prior analysis. To generate these insights, we convert the analysis charts depicted in the prior sections into structured text and query our evaluator LLM to highlight important and actionable trends and insights. Figure 11 illustrates insights generated by QUALEVAL for predictions from DAVINCI-3 model on MBPP, which adeptly highlights model deficiencies for both domains and sub-tasks. For instance, the insights faithfully point out that improvements can be made to the "Tuple Manipulation" and "Number Manipulation" domains as well as the "Algorithmic operations" and "Handling loops and conditionals" sub-tasks. In the next section, we demonstrate how these insights can be leveraged towards precise and targeted model improvement, further validating the efficacy of QUALEVAL.

4.4 Model Improvement via Qualitative Evaluation

We show that QUALEVAL's actionable insights are useful by improving models on a variety of settings on the DialogSum dataset. We leverage insights from QUALEVAL to *precisely* and *consistently* improve the proficiency of a 7 billion parameter Llama 2 model.

Consider a real-world scenario where certain sub-domains are more important. For example, in a toxicity detection dataset, you would expect sub-domains relating to *racial abuse* to have better accuracy than say *politics*. In such a case, a practitioner would want to identify if there are critical sub-domains where the model under-performs, and fix those issues. We consider this scenario for the Llama model where the practitioner is allowed to augment a certain number of instances of their choice to the training set based on the insights. This simulates the scenario where only a certain number of annotated examples can be obtained because of data paucity and cost reasons.

Assume there is a set of sub-domains that the model

Domain sets			Rand. aug.			QUALEVAL aug.			Δ = (QUALEVAL aug. – Rand. aug.)↑			
Dom 1	Dom 2	Dom 3	Dom 1	Dom 2	Dom 3	Dom 1	Dom 2	Dom 3	Dom 1	Dom 2	Dom 3	Overall
Social	✗	✗	27.6	✗	✗	30.0	✗	✗	**2.4**	✗	✗	**2.6**
Leisure	Outdoor	✗	26.6	27.1	✗	29.0	27.7	✗	**2.4**	**0.6**	✗	**3.1**
Food ordering	Hospitality	✗	27.8	28.3	✗	31.5	31.5	✗	**3.7**	**3.2**	✗	**3.6**
Leisure	Food Ordering	Hospitality	26.6	27.8	28.3	32.0	32.0	31.2	**5.4**	**4.2**	**2.9**	**4.1**

Table 1: QUALEVAL consistently increases the performance (ROUGE-L) of the Llama 2 (7 billion parameter) model on DialogSum. QUALEVAL enables practitioners to do targeted model improvement through data augmentation, while keeping the training set size constant. We demonstrate improvements across different sets of domains (with different domains and different numbers of domains) and show consistent and significant improvements on the selected domains along with improvements in overall performance (refer to columns under "Δ"). For instance, augmenting with the "Leisure", "Food ordering", and "Hospitality" domains (last row) leads to an *absolute* overall improvement of 4.1 percentage points.

is underperforming in, as identified by QUALEVAL. QUALEVAL's flexible LP solver finds a set of unsupervised examples belonging to these domains that are then annotated and added to the training instances. We compare with a baseline (Rand. Aug.) that randomly annotates and augments the same number of instances from the unsupervised store. We experiment with different sets of under-performing domains in Table 1 (pertaining to different rows) by fine-tuning the Llama 2 model on the two augmented dataset settings. Additional details are presented in Appendix A.4.

Across different sets of domains (rows), QUALEVAL consistently and significantly increases the proficiency of the selected domains and the overall performance (Table 1). For instance, augmenting with the "Leisure", "Food ordering", and "Hospitality" domains (last row) leads to an improvement of 5.4%, 4.2%, and 2.9% on ROUGE-L on the respective domains and an overall improvement of 4.1% on ROUGE-L, when compared to *Rand. Aug*. Taken together, QUALEVAL empowers practitioners to improve model proficiency with a high degree of *precision* and *control*.

5 Analysis

5.1 Skill usage calibration between ground truth and model answers

While proficiency metrics like pass@k, BLEU, and ROUGE can judge the proficiency of a model, they do not provide insights about skill usage calibration, i.e., whether the model is leveraging the expected subtasks when generating responses. Skill usage calibration is a unique lens to understand model performance, as practitioners can understand if the model generates answers with the expected and intended reasoning.

We quantify the calibration by first identifying the affinity of the ground truth and model-generated answers to different sub-tasks discovered. We then measure the distance between the affinity scores. A smaller distance implies that the model-generated answer is using sub-tasks similar to the ground truth answer, thus exhibiting high skill usage calibration. We explain the exact distance metric used in Appendix A.5.

Figure 6 highlights the correlation between model generations and ground truth responses for the DAVINCI-3 model on the MBPP and DialogSum datasets. A model practitioner can utilize this to understand what sub-tasks are not being used in an intended way and perform an intervention to fix them. For example, on the MBPP dataset, the subtasks about implementing algorithms and handling loops and conditionals have low alignment.

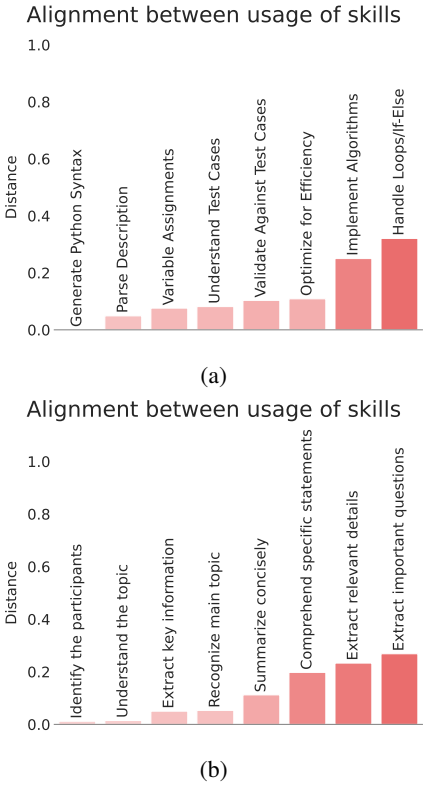


Figure 6: Skill usage calibration between the ground truth and the model generated answer for the DAVINCI-3 model on MBPP (top) and DialogSum (down). A smaller distance implies that the model is using sub-tasks as intended in the ground truth (hence it is better).

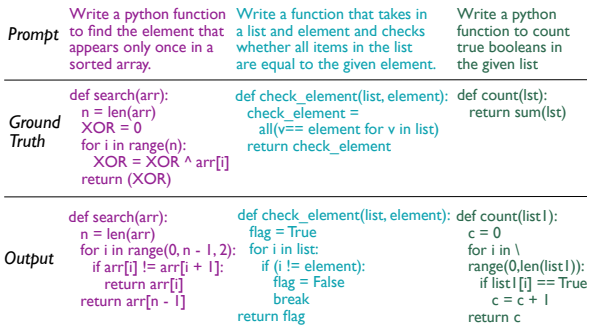


Figure 7: Qualitative samples from the MBPP dataset retrieved through QUALEVAL. We provide an explanation for their prominence in Section 5.2.

5.2 Qualitative Samples

QUALEVAL also allows model developers to extract prominent qualitative examples that can aid in the modeling lifecycle. Given that both in an academic and industry setting, understanding representative instances of ground truth and model-generated answers is important, QUALEVAL automates that process. It *automatically* yields revealing qualitative samples by identifying samples where the affinity scores of the ground truth response and model generation are not aligned.

Figure 7 shows qualitative samples from the MBPP dataset generated by the DAVINCI-3 (left and center) and DAVINCI-2 (right) models. In the first example, the ground truth program uses XOR to test for uniqueness, while the generation uses a loop to check for uniqueness. In the second example, the ground truth program uses an in-built Python function to check equality whereas the model loops through the input to check the condition. These examples further validate the finding in the prior section which suggests that the model is not calibrated for handling loops and conditionals.

Interestingly, the generated output in the final example is a more robust solution than the ground truth. The ground truth solution assumes that the input is a list of booleans, while the model generation can accept any list with any data type.

6 Related Work

Model Debugging/Improvement Prior work has attempted to address the problem of model debugging and improvement. (Zhang et al., 2018) propose to evaluate different pairs of models on separate evaluation splits to understand model behavior. They also generate feature-level importance scores from "symptom" instances provided by humans. (Graliński et al., 2019) introduce a model-agnostic method to find global features that "influence" the model evaluation score, allowing practitioners to exclude problematic features. (Lertvittayakumjorn and Toni, 2021) develop a framework to generate explanations for model predictions to allow humans to give feedback and debug models. (Ribeiro et al., 2020) presents a framework to generate test cases at scale to evaluate model robustness, but constrains the test cases to be generated from simple templates and lexical transformations. (Abid et al., 2022) propose a framework to generate counterfactual explanations for model errors to enable a better understanding of model behavior. (Chen et al., 2023) introduce Self-Debugging, a method to enable a large language model to debug the predicted computer program through few-shot demonstrations. Some other works attempt to find error-prone slices of the data to improve the model (He et al., 2021; Tornede et al., 2023; Paranjape et al., 2022). While these works provide limited insights into model behavior, they often require significant human intervention to understand model behavior and do not provide precise actionable insights for model improvement. Finally, these works are constrained to simple classification and regression tasks or single domains like code generation and do not provide a task-agnostic, fully automated framework for model interpretation and improvement for real-world tasks.

Automatic Evaluation of Machine Learning Models Automatic evaluation metrics, based on lexical overlap, such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005) have helped researchers evaluate and compare models on a variety of language tasks. Recent work has proposed to use machine learning models to evaluate other machine learning models. Methods like (Zhang et al., 2019; Fu et al., 2023; Zhou et al., 2023) use pre-trained language models to evaluate the quality of generated text and therefore rely more on semantics than lexical overlap. While these automated metrics have expedited research progress by eliminating human effort from evaluation, they have limited evaluation to a single scalar metric and therefore fail to provide a holistic and comprehensive understanding of model performance.

Issues with quantitative metrics Multiple studies have pointed out that quantitative metrics are not sufficient to understand the behavior of LLMs and that they are not a good proxy for real-world performance (Liu and Liu, 2008; Novikova et al., 2017; Reiter and Belz, 2009; Liu et al., 2016). While these studies advocate better quantitative metrics, our study proposes a new framework based on *qualitative* evaluation.

7 Conclusion

We propose QUALEVAL, a qualitative evaluation framework that provides a comprehensive way of evaluating models with a keen eye on model improvement. Rather than rely on scalar quantitative metrics that ignore the nuanced behavior of the model, QUALEVAL augments quantitative metrics to test the model thoroughly and provides actionable insights through an interpretable dashboard to improve the model iteratively. We demonstrate that these insights are faithful and lead to up to 15% relative improvement. Our work is the first step towards building a data-scientist in a box.

References

Abubakar Abid, Mert Yuksekgonul, and James Zou. 2022. Meaningfully debugging model mistakes using conceptual counterfactual explanations. In Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pages 66–88. PMLR.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. arXiv preprint arXiv:2108.07732.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023. Teaching large language models to self-debug. arXiv preprint arXiv:2304.05128.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. Dialogsum: A real-life scenario dialogue summarization dataset. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 5062–5074.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. arXiv preprint arXiv:2302.04166.

Filip Graliński, Anna Wróblewska, Tomasz Stanisławek, Kamil Grabowski, and Tomasz Górecki. 2019. GEval: Tool for debugging NLP datasets and models. In Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 254–262, Florence, Italy. Association for Computational Linguistics.

Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. Automl: A survey of the state-of-the-art. Knowledge-Based Systems, 212:106622.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In International Conference on Learning Representations.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In International Conference on Learning Representations.

Piyawat Lertvittayakumjorn and Francesca Toni. 2021. Explanation-based human debugging of NLP models: A survey. Transactions of the Association for Computational Linguistics, 9:1508–1528.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. arXiv preprint arXiv:1603.08023.

Feifan Liu and Yang Liu. 2008. Correlation between rouge and human evaluation of extractive meeting summaries. In Proceedings of ACL-08: HLT, short papers, pages 201–204.

Jekaterina Novikova, Ondrej Dusek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, pages 2241–2252. Association for Computational Linguistics.

OpenAI. 2023. Introducing chatgpt.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Proceedings of the Conference on Health, Inference, and Learning, volume 174 of Proceedings of Machine Learning Research, pages 248–260. PMLR.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Bhargavi Paranjape, Pradeep Dasigi, Vivek Srikumar, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. Agro: Adversarial discovery of error-prone groups for robust optimization. arXiv preprint arXiv:2212.00921.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. Computational Linguistics, 35(4):529–558.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4902–4912.

Alexander Tornede, Difan Deng, Theresa Eimer, Joseph Giovanelli, Aditya Mohan, Tim Ruhkopf, Sarah Segel, Daphne Theodorakopoulos, Tanja Tornede, Henning Wachsmuth, et al. 2023. Automl

in the age of large language models: Current challenges, future opportunities and risks. arXiv preprint arXiv:2306.08107.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Jiawei Zhang, Yang Wang, Piero Molino, Lezhi Li, and David S Ebert. 2018. Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. IEEE transactions on visualization and computer graphics, 25(1):364–373.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In International Conference on Learning Representations.

Shuyan Zhou, Uri Alon, Sumit Agarwal, and Graham Neubig. 2023. Codebertscore: Evaluating code generation with pretrained models of code. arXiv preprint arXiv:2302.05527.

A Appendix

A.1 Limitations

QUALEVAL can faithfully discover relevant sub-tasks and domains and can generate interpretable and actionable dashboards from model predictions. However, we only demonstrate QUALEVAL on diverse language tasks like code generation, summarization, and question answering but do not demonstrate results on multilingual and multi-modal tasks. Our evaluator model is a closed-source model (GPT-3.5), and replacing it with a performant open-source model will make our contribution more accessible to the community. We hope future work will address these challenges and extend QUALEVAL to be an even more general paradigm.

A.2 Ethical Considerations

Our work provides a potent way to ensure that certain tasks performed by data scientists can be automated. While this reduces the burden on them, it is also possible that it reduces the need to have a very large group of them on a certain project. This might have workforce implications. But the intention of the study is to show that with the current LLMs, we can improve evaluation by making it comprehensive.

A.3 Prompts used in QUALEVAL

A.4 Model improvement

We first leverage QUALEVAL's flexible LP solver to generate domain assignments for training samples. We then choose a base set of 250 training samples and leverage the domain assignments to augment the training set by adding 250 additional samples from the training set from up to 3 different domains. Therefore we randomly sample $\frac{250}{N}$ samples from the selected domains, where N is the number of selected domains ($N \leq 3$). We experiment with different sets of domains in Table 1. We then train the off-the-shelf Llama 2 model on these augmented datasets and present both the ROUGE-L scores of the model on the selected domains (refer to "QUALEVAL Aug." columns) and the overall improvement of the ROUGE-L score of the model on the evaluation set (refer to "Δ – Overall" column). For the baseline, we use the same training set but randomly augment the training set with 250 samples (refer to "No Aug." columns).

A.5 Distance metric for skill usage calibration

We measure the distance between the affinity scores by measuring the fraction of samples where the difference between the affinity scores of the generation and the ground truth is greater than 1.

A.6 Example natural language insight

A.7 Dashboards

MBPP

Domain: Given the following examples, What are relevant domains for the following programs? Focus on the example programs BUT be general. Structure the response as a numbered list.

Sub-task: Given the example programs, What are specific ATOMIC sub-tasks a machine learning model need to be competent at for the underlying task? Focus on the example programs BUT be general. [IMPORTANT] Do NOT list the overall task as a subtask and be GENERAL. Structure the response as: Subtask:. Generate a numbered list.

DialogSum

Domain: Given the following conversations, What are relevant domains for the data? Focus on the example data BUT be general. Structure the response as a numbered list.

Sub-task: Given the example conversations, What are specific sub-tasks a machine learning model need to be competent at for the underlying task? Focus on the example data BUT be general. [IMPORTANT] Do NOT list the overall task as a subtask and be GENERAL. Structure the response as: Subtask:. Generate a numbered list.

MMLU (Clinical Knowledge)

Domain: Given the following examples, What are relevant domains for the data? Focus on the example data BUT be general. Structure the response as a numbered list.

Sub-task: Given the example questions and answers on clinical biology, What are the subtasks a machine learning model needs to be competent at to be a good medical assistant. Focus on the example data BUT please be general. [IMPORTANT] Do NOT list the overall task as a subtask and be GENERAL while being GROUNDED in the example data. Structure the response as: Subtask: <subtask>. Generate a numbered list.

Figure 8: Prompt used for discovering attributes across different tasks.

Domain

Given the input to a language model, Rate to what degree the input belong to each of the following domains. Rate on a scale of 1-5, with 5 being compeletely belongs and 1 being not belonging at all. [Important] For each domain, format the output as, [Domain 1: <domain>, Score: <score>, Evidence: <Evidence for score>] [Domain 2: <domain>, Score: <score>, Evidence: <Evidence for score>] [Domain N: <domain>, Score: <score>, Evidence: <Evidence for score>]. [Important] Make sure to include concrete evidence based on the input to JUSTIFY the score. Remember you are an ACCURATE, FAITHFUL, CRITICAL and FAIR judge.

Subtask

Given the input to a language model, Rate to what degree each of the following subtasks are needed to successfully understand and complete the task. Rate on a scale of 1-5, with 5 being very used and 1 being not used at all. [Important] For each subtask, format the output as [Subtask 1: <subtask>, Score: <score>, Evidence: <Evidence for score>] [Subtask 2: <subtask>, Score: <score>, Evidence: <Evidence for score>] [Subtask N: <subtask>, Score: <score>; Evidence: <Evidence for score>]. [IMPORTANT] Do NOT add \n between subtask, score and explanation. [Important] Make sure to include concrete evidence based on the input to JUSTIFY the score. Remember you are an ACCURATE, FAITHFUL, CRITICAL and FAIR judge.

Figure 9: Prompt for scoring attributes.

Insight Generation

System: Given a holistic picture of the performance of a machine learning model, you are asked to summarize the model's overall performance.

Prompt: Given the above information, please write a brief summary highlighting important information. Please be precise and concise but please be comprehensive.

A machine learning model is tasked with the following task: { task instruction }

These are the subtasks/domains for the task: list of subtasks/domains

In the evaluation data, these are the importance scores of the Subtask/Domains: {json.dumps(prior probabilities of subtasks and domains)}

The following scores show how well the model performs on the subtasks/domains:{json.dumps(proficiency scores of subtasks and domains)}

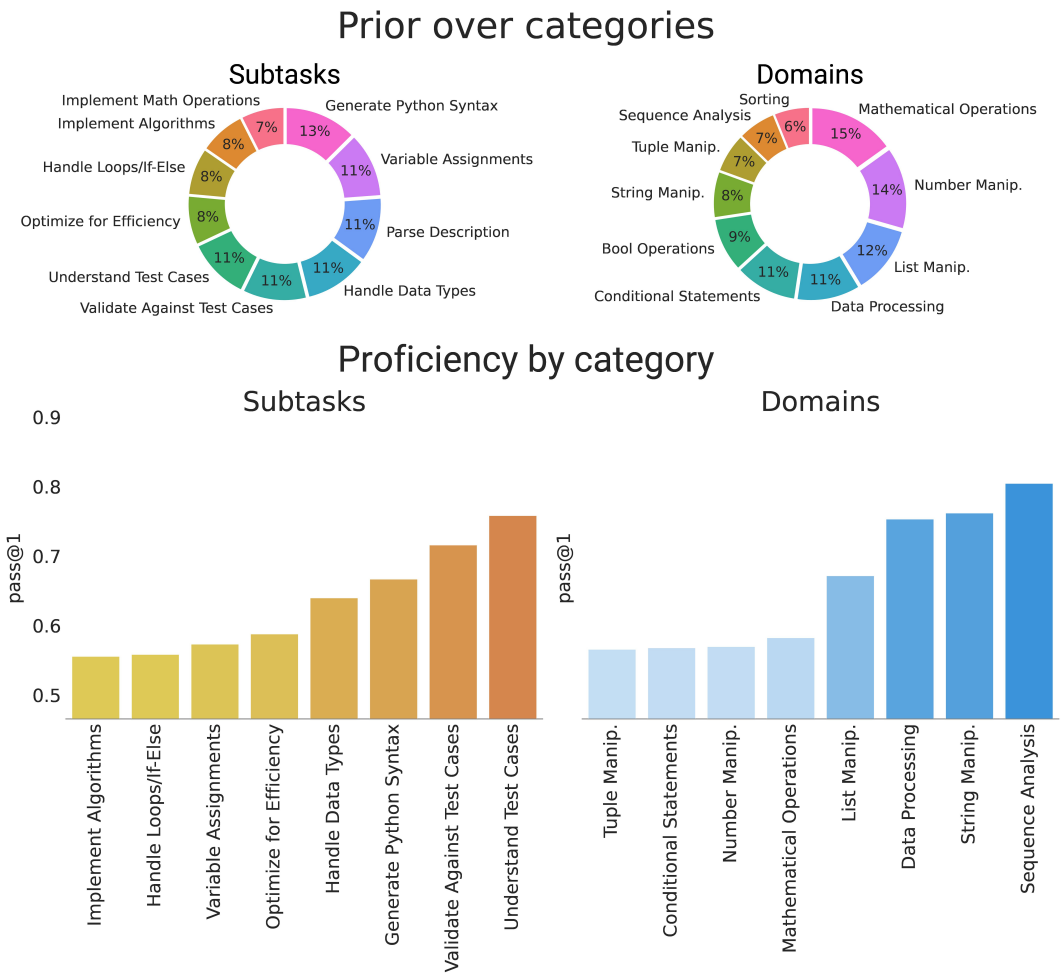
The following distance demonstrates how much the domains/subtasks are actually used for generating the output when they are requried to generate the input. Therefore, a low distance implies that the model is utilizing the category when it needs to: {json.dumps(correlation scores of category)}. [Important] Lower distance implies the category is leveraged when it needs to be used.

Figure 10: Prompt for generating insights.

QUALEVAL Insights

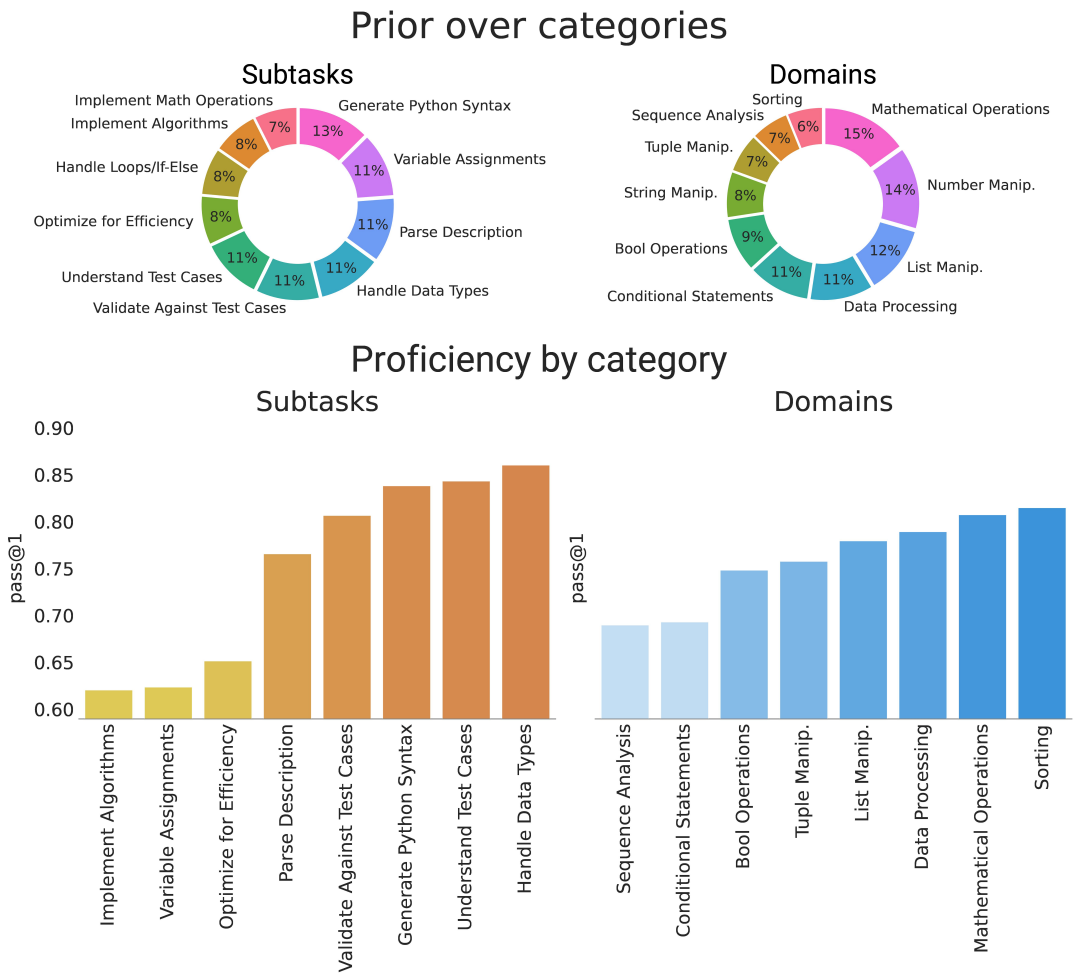
The machine learning model performs well on various subtasks, with the highest scores in "Understand test cases" and "Validate against test cases". It also excels in "Generate Python syntax" and "Manage variable assignments and data manipulation". However, it could improve in "Implement algorithmic operations" and "Handling loops and conditionals". The model effectively utilizes the subtasks when generating the output, particularly in "Generate Python syntax" and "Implement mathematical operations". In terms of domains, it performs strongly in "Sequence Analysis" and "String Manipulation", while improvements can be made in "Tuple Manipulation" and "Number Manipulation". Overall, the model demonstrates proficiency in understanding the requirements and generating accurate Python code, with potential for further enhancements in specific areas.

Figure 11: Natural language insights generated by QUALEVAL for the *davinci-2* model on MBPP.



The machine learning model performs well on various subtasks, with the highest scores in "Understand test cases" and "Validate against test cases". It also excels in "Generate Python syntax" and "Manage variable assignments and data manipulation". However, it could improve in "Implement algorithmic operations" and "Handling loops and conditionals". The model effectively utilizes the subtasks when generating the output, particularly in "Generate Python syntax" and "Implement mathematical operations". In terms of domains, it performs strongly in "Sequence Analysis" and "String Manipulation", while improvements can be made in "Tuple Manipulation" and "Number Manipulation". Overall, the model demonstrates proficiency in understanding the requirements and generating accurate Python code, with potential for further enhancements in specific areas.

Figure 12: MBPP - DAVINCI-2



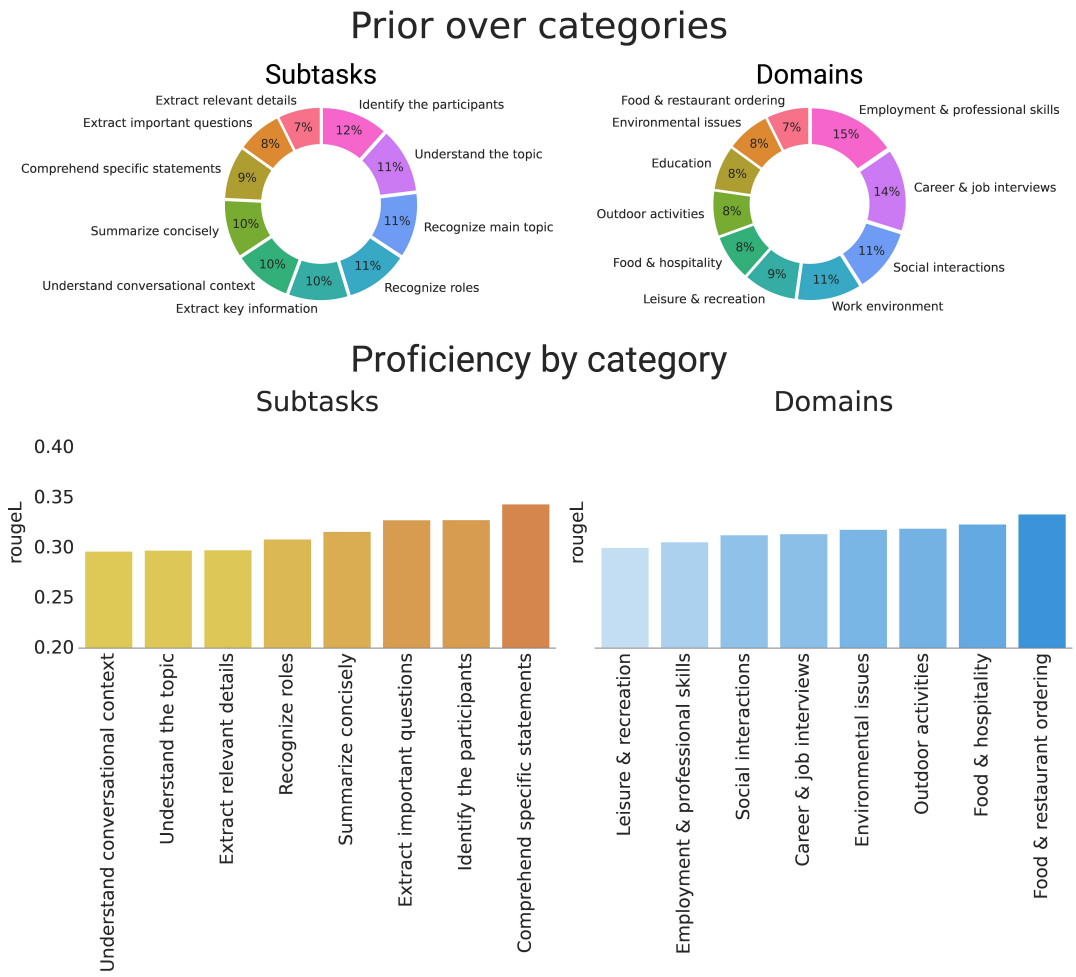
The machine learning model displays strong performance across various subtasks and domains. It accurately implements mathematical and algorithmic operations, handles loops and conditionals, and optimizes for efficiency and readability. The model demonstrates a good understanding of test cases and effectively validates against them. It effectively manages variable assignments and data manipulation and generates Python syntax with precision. Additionally, it shows a strong ability to parse natural language descriptions.

The model performs well in different domains, such as sorting, sequence analysis, tuple manipulation, string manipulation, boolean operations, conditional statements, data processing, list manipulation, number manipulation, and mathematical operations.

Furthermore, the model effectively utilizes subtasks when needed, as indicated by low distances between the required and actual usage of subtasks. This highlights its ability to leverage the necessary subtasks in generating the desired outputs.

Overall, the model exhibits a comprehensive understanding of the task and performs with high accuracy and efficiency, making it a reliable tool for generating Python functions from natural language descriptions and associated test cases.

Figure 13: MBPP - DAVINCI-3

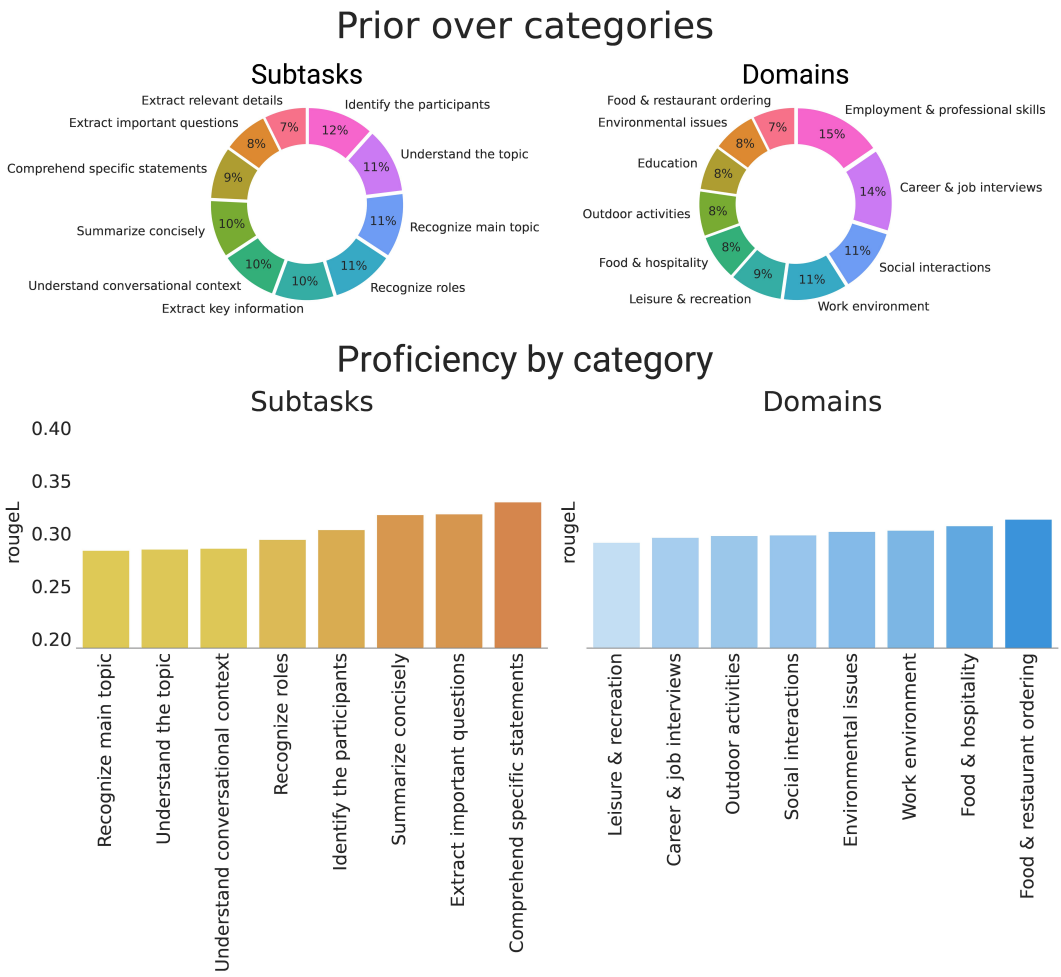


The machine learning model performs well in comprehending specific statements or questions and recognizing the roles and relationships of the speakers, with scores of 0.342 and 0.307 respectively. It also demonstrates good understanding of the main topic of conversation and the conversational context, with scores of 0.313 and 0.295 respectively. However, it slightly struggles in extracting relevant details and summarizing the conversation concisely, with scores of 0.296 and 0.315 respectively.

In terms of domains, the model performs best in the Food and restaurant ordering domain with a score of 0.332, followed closely by Environmental issues and pollution with a score of 0.317. It performs relatively weaker in the Leisure and recreation domain with a score of 0.299.

Overall, the model shows good utilization of subtasks, particularly in identifying the participants in the conversation and understanding the topic of discussion, with very low distances of 0.004 and 0.021 respectively. However, it could improve in efficiently utilizing the subtasks of extracting relevant details and extracting important information and questions, with distances of 0.260 and 0.268 respectively.

Figure 14: DialogSum - DAVINCI-2



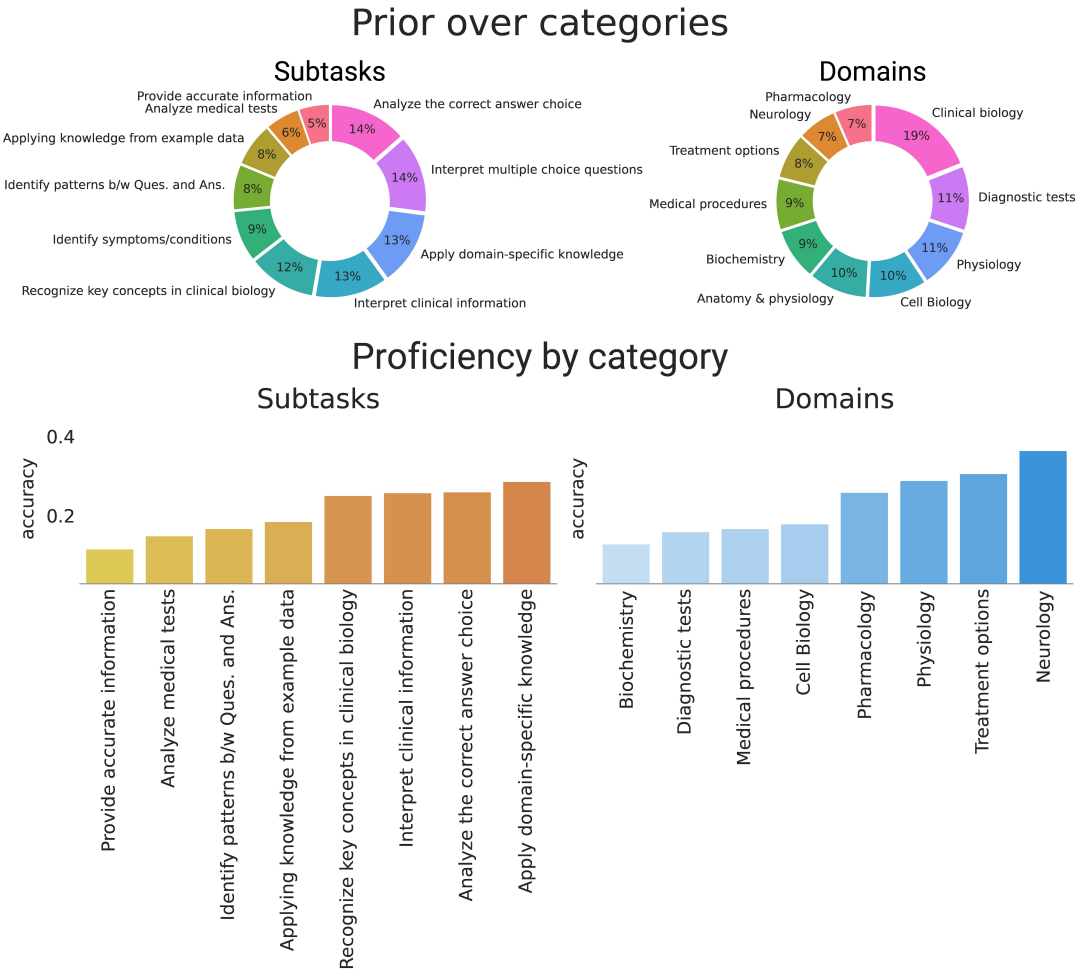
The machine learning model performs well in recognizing and understanding the main topic of conversation, comprehending specific statements or questions, and extracting key information or important details. It also excels in summarizing conversations concisely and understanding the conversational context. Additionally, it demonstrates good performance in identifying participants in the conversation and recognizing the roles and relationships of the speakers.

When it comes to the domains, the model performs relatively well in leisure and recreation, career and job interviews, and outdoor activities and sports. It also shows satisfactory performance in social interactions and personal relationships, employment and professional skills, and education.

Furthermore, the model effectively utilizes the subtasks when required, particularly in identifying participants and understanding the topic of discussion. It demonstrates lower distances, indicating the subtasks are leveraged as needed.

Overall, the model achieves strong performance in various aspects of conversation understanding and has a good grasp of different domains.

Figure 15: DialogSum - DAVINCI-3

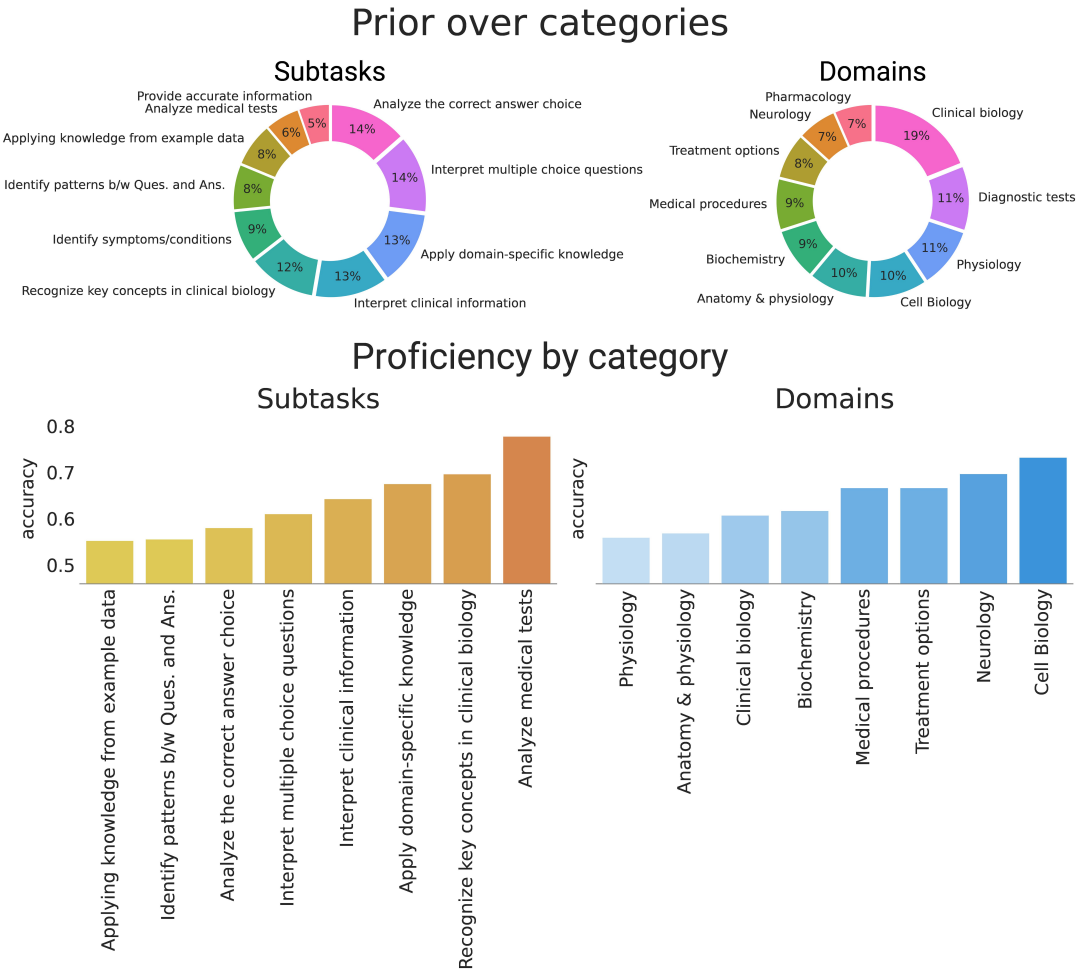


The machine learning model has been evaluated on various subtasks and domains related to clinical biology. Among the subtasks, the model performs relatively well in understanding and interpreting multiple choice questions, recognizing key terms and concepts in clinical biology, and applying domain-specific knowledge to select the most appropriate answer choice. However, it needs improvement in providing accurate and relevant information to healthcare professionals and patients, analyzing and processing medical test results, and identifying patterns and relationships between questions and answers.

In terms of domains, the model shows higher performance in physiology, treatment options, and pharmacology. On the other hand, it performs comparatively lower in biochemistry, diagnostic tests, and medical procedures and interventions. Notably, clinical biology has the highest importance score among the domains.

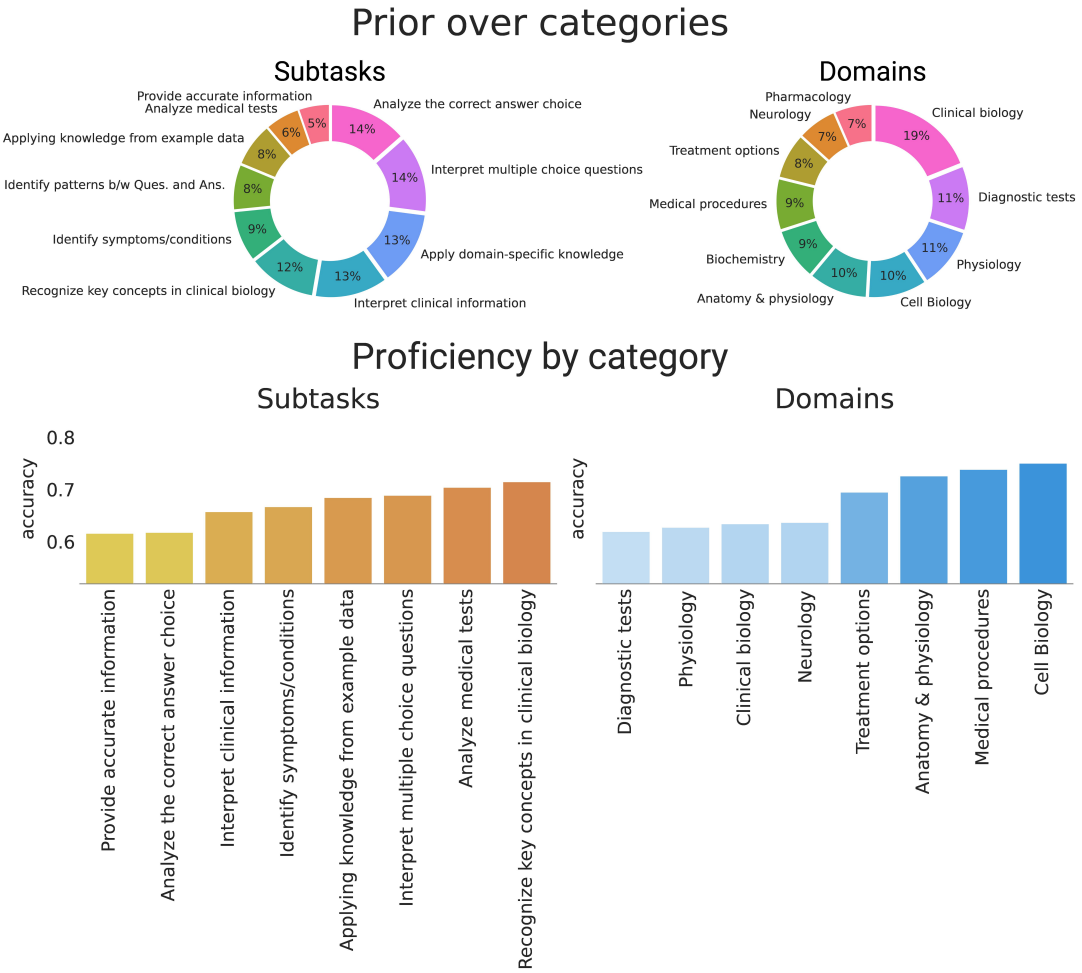
Overall, the model demonstrates a good understanding and interpretation of clinical information, but there is room for improvement in specific subtasks and domains to enhance its performance.

Figure 16: MMLU (Clinical Knowledge) - CURIE



The machine learning model performed well on multiple subtasks, with the highest scores in analyzing and selecting the correct answer choice, understanding and interpreting multiple choice questions, and providing accurate and relevant information. It also excelled in applying domain-specific knowledge and recognizing key terms and concepts in clinical biology. The model showed good performance in retaining and applying knowledge from example data, identifying patterns and relationships between questions and answers, and interpreting clinical information. In terms of domains, the model performed strongly in clinical biology, diagnostic tests, pharmacology, and medical procedures and interventions. However, it had lower performance in neurology and physiology. Overall, the model demonstrated a solid understanding of clinical biology and was able to analyze and select the correct answer choice effectively.

Figure 17: MMLU (Clinical Knowledge) - DAVINCI-2



The machine learning model performs well across different subtasks and domains in clinical biology. It excels in understanding and interpreting multiple choice questions, analyzing and selecting the correct answer choice, and applying domain-specific knowledge. Recognizing key terms and concepts, as well as identifying patterns and relationships between questions and answers, are also strong areas for the model. It demonstrates good performance in understanding and interpreting clinical information and identifying symptoms, conditions, and diseases. The model performs best in analyzing and processing medical test results. Among the domains, it performs exceptionally well in cell biology, physiology, and medical procedures and interventions. The model also shows promising performance in anatomy and physiology, diagnostic tests, and treatment options.

Figure 18: MMLU (Clinical Knowledge) - DAVINCI-3