An Evaluation of Classifiers for Mapping Generative LLM Responses to Answer Options of Multiple-choice Questionnaires

Alisea Stroligo
University of Konstanz
alisea.stroligo@uni-konstanz.de

Anna Shamray
University of Konstanz
anna.shamray@uni-konstanz.de

Julian Schelb
University of Konstanz
julian.schelb@uni-konstanz.de

Andreas Spitz
University of Konstanz
andreas.spitz@uni-konstanz.de

Abstract

The use of large language models (LLMs) for generating responses to multiple-choice style questionnaires that were originally intended to be answered by humans is often a helpful or even necessary task, for example in persona simulation or during LLM alignment. Although the input and output versatility of generative LLMs is beneficial when adapting such questionnaires to machine use, it can be detrimental when mapping the generated text back to a closed set of possible answer options for evaluation or scoring. In this paper, we investigate the performance of smaller models for the classification of LLM outputs into the available answer options of multiple-choice questionnaires. We consider fine-tuned encoder-transformers as well as a rule-based approach on three datasets with differing answer option complexity. Surprisingly, we find that the best-performing neural approach still underperforms in comparison to our rule-based baseline, indicating that simple pattern-matching of answer options against LLM outputs might still be the most competitive solution for cleaning LLM responses to multiple-choice questionnaires.

1 Introduction

Large language models are increasingly being used to simulate realistic human-aligned responses to questionnaires originally intended for humans in so-called persona simulation (e.g., see Aher et al., 2023; Argyle et al., 2023; Horton, 2023; Lu and Wang, 2024), or to assess the characteristics of the models themselves (e.g., see Binz and Schulz, 2023; Bodroža et al., 2024; Li et al., 2024; Lu et al., 2023; Miotto et al., 2022; Pellert et al., 2024; Serapio-García et al., 2025) in the emerging field of machine psychology (Hagendorff et al., 2024). A commonly used tool in such settings are multiple-choice questionnaires with a fixed set of answer options (also called closed-ended questions), which also occur frequently when evaluating the natural language understanding capabilities of LLMs with collective evaluation benchmarks such as MMLU (Hendrycks et al., 2020), SuperGLUE (Wang et al., 2018), and BIG-bench (Ghazal et al., 2013), making closed-ended questions one of the simplest and most commonly used answer formats.

For generative LLMs, simulating human respondents or generating a large number of responses to a closed-ended question is as simple as prompting the LLM with a question and the corresponding set of answer options from which to choose. However, given the variability and noise in these outputs, one is then faced with the task of mapping LLM responses back to the set of answer options, which is non-trivial due to LLM verbosity, hallucination, or a model's failure to comprehend the question. In small-scale experiments, such responses may be annotated by human experts, but such an approach is laborious and increasingly impractical with growing scale – rendering automated matching of LLM outputs to the set of answer options the most feasible solution for cleaning LLM responses. While constrained decoding has been proposed as a solution for directly mapping token probabilities to answer options, this has been found to be problematic (Wang et al., 2024a). Likewise, one might adapt generative LLM-as-a-judge approaches used frequently for scoring open-ended questions (Li et al., 2025), but would then struggle with the recursive issue of needing to clean the judge's outputs.

Contributions. To address this problem, we investigate dependable, generalizable, and domain-agnostic approaches for mapping noisy, LLM-generated questionnaire responses to a discrete range of answer options. In particular, we

(i) propose four variations of a simple general-purpose classifier architecture that can be implemented by fine-tuning encoder-based models, which we evaluate against a rule-based pattern-matching baseline;

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 4: Student Research Workshop, pages 861–880

(*ii*) create a dataset of manually annotated responses generated by a variety of state-of-the-art LLMs to multiple-choice questions from three sources with differing complexity of answer options for evaluating the classifiers.

2 Related Work

Text classification is a central task in natural language processing (NLP) and has a wide range of applications, including question answering (QA). Historically popular choices of classification algorithms have included Naïve Bayes algorithms, Support Vector Machines (SVM), K-Nearest Neighbor (KNN) algorithms, Gradient Boosting Trees, or Random Forests. In recent years, a growing body of literature has developed around large language models for text classification (Fields et al., 2024; Gasparetto et al., 2022; Kostina et al., 2025; Li et al., 2022; Minaee et al., 2022), and remarkable results have been achieved with these models (Cunha et al., 2025; Kaliyar et al., 2021; Sun et al., 2023; Zhang et al., 2025), rendering traditional models all but obsolete. However, while generative LLMs can potentially offer a single versatile solution to text classification, they are time- and resource-intensive – for an in-depth exploration of the trade-off between performance improvements and resource- and time-requirements, see Kostina et al. (2025). Furthermore, despite achieving state-of-the-art results, these classifiers also show significant limitations in their applicability, as recently demonstrated by Vajjala and Shimangaud (2025) and Xu et al. (2024).

In resource-constrained settings or for large amounts of data to classify, smaller language models tend to offer better performance-to-time trade-offs. In particular, encoder-architectures such as the one proposed for BERT (Devlin et al., 2018) are designed to be fine-tuned for text classification tasks. Recent work from Gweon and Schonlau (2023) and Schonlau et al. (2023) finds that using a BERT-based model is preferable for automatically classifying open-ended questions than non-pretrained models. For classifying responses to closed-ended questions, we find that the task of extractive question answering within the two-sentence structure of BERT-style models corresponds well to the task of classifying whether a generative LLM response contains a valid answer option, and we therefore focus on such model architectures. The only directly related work with a similar approach of which we are aware is by Schelb et al. (2025), from which we take inspiration for the classifier design and use one of their annotated data sets as a starting point for our experiments.

3 Problem Statement

Closed-ended questions require LLMs to answer by choosing one out of a finite number of available answer options. However, even state-of-the art generative LLMs often produce responses that include content beyond the requested answer, such as reasoning explanations, source citations, answer refusals, etc. In order to use such LLM responses in downstream analyses, it is necessary to match the responses to the answer options that are associated with the question. The set of possible responses can be separated into two classes: valid responses containing a single answer option and invalid responses containing no or multiple answer options.

Answer Option Class. All LLM responses that correspond to one of the provided answer options fall into this class, and a suitable classifier should be able to determine which answer option is the best match. For example, consider the following question from the Regulatory Focus Questionnaire (Higgins et al., 2001):

> **Input prompt:** `Choose your answer to the question "Do you often do well at things that you try?" by choosing from the following list of options:`
> `1. never or seldom`
> `2.`
> `3. sometimes`
> `4.`
> `5. very often.`
> **LLM Response:** `The best response is "sometimes 3."`

While the response does not match an answer option perfectly and contains added reasoning, it matches the valid answer option `3. sometimes` and should be classified accordingly.

[None] Class. In contrast, the [None] class is comprised of responses that should not be matched to an answer option and consists of two possible cases: *not present* and *inconclusive*.

In some responses, a valid answer option might not be present. For example, consider the following response to the question from the above example:

LLM Response: `As an AI language model, I cannot help with that.`

Likewise, some generated outputs may be inconclusive due to ambiguity, for example when the LLM generates responses containing multiple answer options or a mixture of fragments from multiple answer options. For example, consider the following responses:

LLM Response: `The correct answer is: 3. sometimes and 1. never or seldom`
LLM Response: `1. sometimes`

In the following, we discuss the design of classifiers that are capable of deciding which answer options a generated LLM response matches best or whether it should be classified as [None].

4 Classifier Design

Given the wide range of styles that answer options for multiple-choice questions may take, ranging from numbered Likert-scale answer options to trivia, our goal is the design of a generalizable, domain- and data-agnostic classifier. Here, the underlying intuition utilizes the 2-input structure that is common to pre-trained encoder-transformer models. Specifically, we fine-tune the model to recognize patterns: whether the input that corresponds to the generative LLM response semantically matches the input that corresponds to a given answer option. Formally, we therefore define answer classification as the task of mapping an LLM output to one of n predefined answer options for each question item. Based on this intuition, we consider four different designs for a neural classifier that can be implemented using any BERT-variant model, as well as a rule-based baseline. For a schematic view of the base classifier design, see Figure 1.

4.1 Rule-based Classifier (RbC)

As a baseline, we consider a rule-based classifier leveraging string-matching. The classifier first tokenizes the answer option into components (e.g., `3. sometimes` would be split into the two tokens `3` and `sometimes`). After removing noise from the response text and preprocessing it by lower-casing and removing punctuation, the classifier checks the response for occurrences of each answer option token by matching whole tokens. For answer options consisting of multiple tokens, the occurrence of each distinct token scores as a fractional

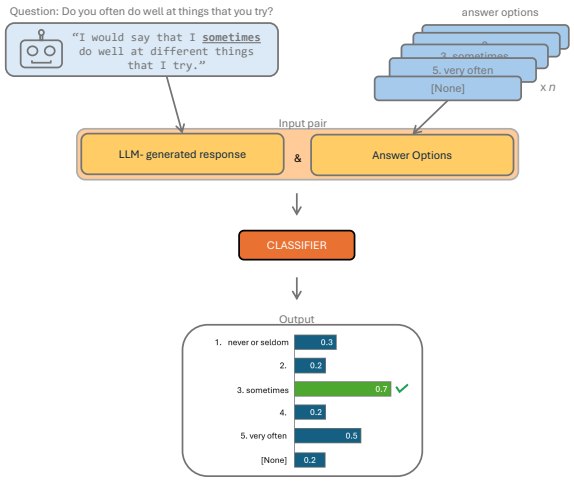

Figure 1: Schematic overview of the classifier design: to classify one LLM response, a fine-tuned binary classifier receives an input pair (answer option - LLM response) for each possible answer option, including a [None] answer option representing an invalid response. Each input pair is assigned a probability value, from which the answer option with the highest probability value is selected as the most likely answer.

value towards this answer option (such that the maximum score for each answer option is 1, independent of its length), and then selects the highest scoring answer option. If no match is found or two options tie for the highest score, the response is classified as [None]. We note that alternative rule-based approaches are feasible (e.g., fuzzy-matching, entailment-based scoring, or IR-style retrieval scoring), but here use the simplest rule-based approach that can still perform closed-ended answer classification.

4.2 Model-based Classifiers

The model-based classifiers are BERT-variant models trained to classify noisy LLM responses vs. one of a set of multiple-choice options as a binary 1-vs-all classification task, with an added class for [None]. To facilitate the prediction of [None] Class responses, we explore two alternatives: entropy-based classification and a traditional multi-class approach with a dedicated [None] class.

4.2.1 Entropy-based Classifiers

To decide whether a response corresponds to an answer option in the Answer Option Class, the entropy-based models work as a typical 1-vs-all classifier by taking pairs of answer options together with the LLM response and computing a probability for each individual answer option-response pair.

The answer option with the highest probability is then predicted as output. However, to predict responses in the [None] Class, the model utilizes an entropy-based approach, in which we compute the entropy over the distribution of answer options probabilities. If the entropy is sufficiently high to cross an empirically determined threshold, the model is assumed to be unable to determine a suitable class and [None] is predicted instead. This approach ensures flexibility of a single fine-tuned model with an arbitrary number of answer options. For further details on the entropy calculation, see Appendix B.

Based on the above intuition, we consider three design variants utilizing an entropy threshold that only differ in their approach to including [None] Class answers in their training data.

Original Entropy Model (EM-O). The base model we consider here was originally proposed by Schelb et al. (2025) for classifying Likert-scale answer options in psychometric questionnaires. The defining feature of this classifier is the absence of any [None] Class instances in the training data.

Entropy Model Variant 1 (EM-1). As an alternative to the base entropy approach, we consider a variant that adds instances from the [None] class to the training data. Specifically, [None] Class instances occur as negative samples for the Answer Option Class, meaning that [None] Class answers are added to the training data for each answer option with an associated binary classification label of 0. For example, the training data would include an instance ⟨`As an AI language model, I cannot help with that | 3. sometimes | 0`⟩.

Entropy Model Variant 2 (EM-2). As a further addition, we consider a model that also includes negative examples for the [None] Class to help the model distinguish it from positive instances of the Answer Option Class. To this end, instances of [None] Class responses are added with a [None] answer option. For example, the training data could include an instance ⟨`The evidence strongly supports 3.sometimes | [None] | 0`⟩.

4.2.2 No-Entropy Model (NEM)

As a more traditional setup, we also consider a no-entropy design in which we simply model the task as a binary 1-vs-all classifier with the addition of a dedicated [None] label, for which we include both positive and negative instances in the training data. Therefore, labels for instances of the [None] Class can be predicted directly without need for an entropy threshold.

Effectively, for each classifier design, we progressively expand the included training data and the resulting classifier versions differ in the degree to which the [None] Class answer is included during training. For a schematic overview of the training data composition of each classifier see Appendix Table 4. For further details on the model designs, see Appendix B.

5 Data

To develop a versatile classifier capable of accommodating a wide range of diverse questionnaires, we consider a variety of formats for closed-ended questions. Specifically, we follow the traditional classification of question formats by Stevens (1946) into measurement scales based on data type: nominal, ordinal, interval, and ratio. In our selection of dataset and the generation of training data, we therefore consider these formats as possible LLM response types that the classifiers have to handle. For a detailed discussion, see Appendix A.

5.1 Training Data Generation

To train the classifiers, we require annotated LLM responses for several multiple-choice questionnaires covering the two classes we identified in Section 3: the Answer Option Class representing LLM responses that correspond to a single valid answer option for a multiple-choice question, and the [None] Class containing ambiguous (i.e., *inconclusive*) responses or those matching no answer options (i.e., *not present*). For both cases, we generate synthetic data by utilizing a template-filling approach in which we insert answer options into templates simulating (slightly) verbose LLM responses to a question from a questionnaire. The Answer Option Class templates are explicitly handcrafted to resemble common outputs from generative LLMs. Using templates for LLM responses allows for better control of the training data and easier implementation of modifications, this also reduces the likelihood of unexpected behavior emerging from the use of actual outputs in the training data, which could introduce excessive noise. The [None] Class training data contains also real [None] Class responses generated by LLMs, since answers that do not contain any answer option and can therefore be completely unrelated to the questionnaire are more

difficult to reduce to common templates, given the wide variety of responses this set can include.

Answer Option Class Data. To generate data for the Answer Option Class, we utilize a set of 67 handcrafted templates from Schelb et al. (2025) that were created with the aim of resembling outputs from LLMs answering the Regulatory Focus Questionnaire (Higgins et al., 2001). For example, a template might be `The evidence strongly supports ⟨answer option⟩`, where `⟨answer option⟩` can be filled with an answer option such that we know the correct label of the generated response. The answer options are kept the same as in Schelb et al. (2025) since the Regulatory Focus Questionnaire conveniently provides a number of answer options wide enough to create variation in the templates (i.e., numerical/non-numerical/mixed, with labeling/ without labeling, etc.), but limited enough to be able during training to evaluate classification performance for each answer option separately. In addition to the handcrafted templates, we also generate up to 20 paraphrased variations for each template using Llama 3.1 70B (Dubey et al., 2024). To ensure data quality and prevent the rephrasing model from straying too far, we compute Sentence-BERT embeddings (Reimers and Gurevych, 2019) for the original template and the rephrased version and discard rephrases for which the cosine similarity with the original lies in the bottom quartile.

[None] Class Data. For the [None] Class, we instead generate instances in which the presented answer options are *inconclusive* (i.e, ambiguous) or *not present*. Inconclusive responses are generated in the same way as for the Answer Option Class samples by populating handcrafted templates with answer options. However, we use templates that are filled with two answer options, such as `I am not sure, the answer could be ⟨answer option 1⟩ or ⟨answer option 2⟩`. Due to the large number of combinations for answer options, we did not paraphrase templates in this category.

To generate instances containing no valid answer options (not present responses), we reuse the discarded paraphrases from the Answer Option Class with a cosine similarity in the bottom 10th percentile. Furthermore, we prompt three generative LLMs (Qwen 2.5 3B, Llama 3.1 8B and 70B) with incomplete instructions by providing a question from a questionnaire without providing any answer options.

Data Labeling. To generate labels for the training data in both classes, we pair the filled templates with (in)correct answer options and assign the corresponding binary label, depending on whether the answer option used in the response (input 1) matches the answer option (input 2). For example, a positive training example is `⟨I think 3. sometimes | 3. sometimes | 1⟩`, while a negative one is `⟨It is 3. sometimes | 2. | 0⟩`.

We apply this generation scheme to each of the three data sources that we discuss in the following section. In each case, we sample the training dataset uniformly across available answer options to guarantee a balanced number of training samples per answer option. For further details on the training dataset generation, see Appendix A.2.

5.2 Benchmark Data Generation

In order to evaluate the performance of the classifier variants, we generated and annotated real LLM responses to questions from three questionnaire datasets that we chose to represent classification scenarios of increasing difficulty. The Regulatory Focus Questionnaire (RFQ) is a single-task and single answer-type questionnaire (Higgins et al., 2001), the AI2 Reasoning Challenge (ARC) is single-task but multi-answer type (Clark et al., 2018), while the Measuring Massive Multitask Language Understanding Pro Task Dataset (MMLU-Pro) is both multi-task and multi-answer type (Wang et al., 2024b). Between the datasets we included all four answer types (nominal, ordinal, interval and ratio) as well as formats (numeric-only, non numeric-only, and mixed).

We generated answers to the questions from each of the datasets by prompting 5 to 8 different generative LLMs. The LLM responses were then annotated independently by four annotators, each with a graduate education in NLP. Answers with differing annotations were discarded (due to the ease of this task for human annotators, this occurred in less than 10% of cases). For further details on the data creation and annotation, see Appendix A.5.

RFQ Data. For the RFQ data, we expand the annotated dataset from Schelb et al. (2025). Overall, we collect and annotate 1068 responses to the RFQ, generated by prompting eight different models: Qwen 2.5 0.5B and 7B and 32B and 72B (Yang et al., 2024), Llama 3.1 8B and 70B (Dubey et al., 2024), Zephyr 7 (Tunstall et al., 2023), and Gemini 3 Pro (Team et al., 2025). Each model was

Dataset	Task Type	Answer Type	# Answer Options	Size (# samples)	% Answer Option / [None] Class
RFQ	Single-task (Psychometric Test)	Single-type (Likert-scale)	5	1068	90% / 10%
ARC	Single-task (Domain-Specific QA)	Multi-type	4	481	97% / 3%
MMLU-Pro	Multi-task	Multi-type	2-10	498	78% / 22%

Table 1: Overview of the three evaluation datasets and their main characteristics.

prompted to answer the RFQ question items with three different prompt variants (see Appendix A.5). The RFQ answer options are ordinal and consist of 5-point Likert-scales – for an example question with answer options, see Section 3.

ARC Data. Our second dataset consists of 481 LLM answers to questions from ARC (Clark et al., 2018), which is a corpus of science questions, each with four possible answer options including non-ordinal answers. To increase variation in the answer formats, we randomly assign either numerical (i.e., 1., 2., 3., 4.), alphabetical (i.e., A., B., C., D.) or no answer class labels to the answer options for each question item. We prompted five different answering models to each answer a randomly sampled subset of 100 questions from ARC, namely Qwen 2.5 72B, Llama 3.1 70B, Gemma 2 9B (Team et al., 2024), Zephyr 7B and Gemini 3 Pro. An example question from the ARC data is:

> **Question:** A signal from the brain to a muscle in the arm is transmitted by which structures?
> **Answer Options:** A. sensory neurons, B. interneurons, C. motor neurons, D. mechanoreceptor neurons.

MMLU-Pro Data. For the third dataset, we use 498 answers to a modified subset of the Measuring Massive Multitask Language Understanding Pro Task (Wang et al., 2024b) dataset. This dataset is the most complex and comprised of different answer types and answer formats (only-numerical, non-numerical, and mixed). We sample the data to ensure a broad range in the number of answer options per question, ranging from 2 to 10. Similar to ARC, the answer options are additionally randomly assigned numerical, alphabetical or no answer class labels. We then prompt five different LLMs to respond to approximately 100 randomly sampled questions: Qwen 2.5 72B, Llama 3.1 70B, Gemma 2 9B, Mixtral 8x7B (Jiang et al., 2024),

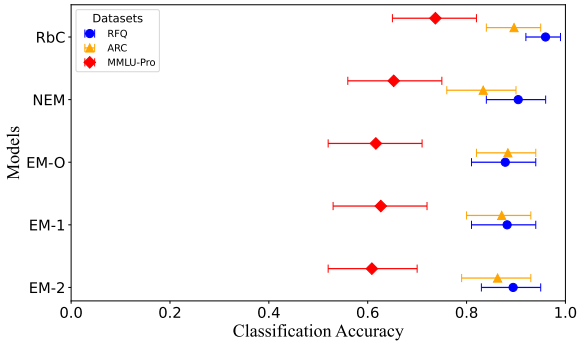

Figure 2: Accuracy scores for the classifier designs on all three datasets. Error bars represent the 95% confidence interval over 100 bootstrap samples.

and Gemini 3 Pro. An example question from the MMLU-Pro dataset is:

> **Question:** What is the (approximate) value of lemon juice on the pH scale?
> **Answer Options:** 1, 0, 9, 2, 10, 12, 14, 5, 7, 8.

An overview of the resulting manually annotated evaluation data is provided in Table 1.

6 Experiments

To evaluate the proposed models, we conduct two sets of experiments: an evaluation of classifier design performance, and an investigation into the impact of modeling choices.

6.1 Experiment 1: Classifier Comparison

We evaluate all classifier designs on each of the three datasets to evaluate the classifiers' versatility and robustness across questionnaires with differing answer types, answer formats, answer labeling, domain, and LLM output quality.

Setup. We implement all neural classifier designs using a RoBERTa model (Liu et al., 2019), which we train for 1 training epoch using 1:3 positive-to-negative sample ratio. We measure classification

accuracy, precision, recall and F1-scores as well as training-times for each classifier.

Results. In Figure 2, we show the performance of all classifiers on each of the three datasets (for the complete results, see Appendix Table 7). Surprisingly, we find the rule-based classifier to have the best performance overall as well as the best performance across all metrics (see Appendix Table 6), even on datasets with the most complex answer options. The performance of the neural classifiers trails closely with no significant difference between the designs – NEM performs slightly better on MMLU than the entropy-based designs, but worse on ARC.

When considering runtimes (see Appendix Table 6), the entropy-based design train substantially faster than the NEM design by a factor of 2, while inference times are negligible across all models – yet still orders of magnitude above the rule-based classifier, which also has no training time.

Qualitative Analysis. Upon manual inspection of the results, we find that the entropy-based designs notably fail to classify any [None] Class answers for the ARC and MMLU-Pro datasets. A possible explanation are the experimentally determined entropy thresholds (see Appendix Table 5) for [None] Class answer selection for these data sets, which are potentially too high. Notably, we find that the generative LLMs produced responses with increased verbosity and proportion of [None] Class answers for the more complex datasets (see Appendix Table 10). This is also reflected in the graded classification performances of all tested classifiers (see Figure 2), which indicates that, as expected by design, the RFQ was the easiest dataset to classify and the MMLU-Pro the most difficult. All classifier versions performed generally worse on more verbose answers, regardless of model size or architecture, and more often underperformed on responses generated by smaller LLMs (see Appendix Table 10).

6.2 Experiment 2: Parameter Sensitivity

We further assess the impact of design choices on the performance of the neural classifier designs. Since all performed comparably in Experiment 1, we only use one model-based classifier design in the following experiments. We choose the NEM design, as overall it was the best performing model-based classifier from Experiment 1.

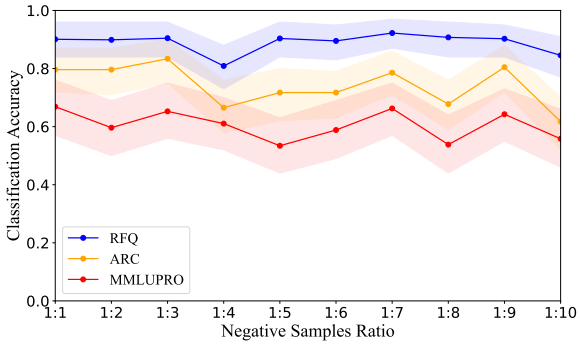

Figure 3: NEM classification accuracies vs. negative sample ratio in the training data. Shaded areas represent the 95% confidence interval on 100 bootstrap samples.

Proportion [None]	Class Size (# samples)	RFQ	ARC	MMLU -Pro
1:1	792	**0.90**	0.83	0.65
1:2	1584	0.89	**0.88**	**0.73**
1:3	2376	0.84	0.82	0.64

Table 2: Training dataset characteristics and NEM classifier accuracy for each of the tested [None] Class proportion variations.

6.2.1 Negative Sample Ratio

Setup. Based on the intuition that in a 1-vs-all setting, negative classifications are more common, we test ten different proportions of negative samples in the training dataset by fine-tuning the classifier on splits ranging from 1 to 10 negative samples per positive sample. For each of the resulting classifiers, we compute the classification accuracy on each of the three benchmark datasets.

Results. As shown in Figure 3 (for details see Appendix Table 8), varying the negative sample ratio does not yield significant variations in the classifier's performance. The 1:3 ratio chosen for Experiment 1 achieves comparable or superior performances to higher ratios across all three datasets. Furthermore, this choice of ratio results in a smaller training dataset than higher ratios and therefore reduces training overhead.

6.2.2 Proportion of [None] Class Data

Setup. We also experimented with the proportion of training data from the [None] Class in relation to the Answer Option Class by training classifiers on proportions of 1:1, 1:2, 1:3 of data from the Answer Option Class compared to the [None] Class.

Results. As shown in Table 2, varying the amount of training data from the [None] Class

Model	Size	RFQ	ARC	MMLU -Pro
ALBERT	12M	0.78	0.51	0.43
DistilBERT	66M	0.87	0.44	0.48
DistilRoBERTa	83M	**0.96**	0.60	0.59
RoBERTa	125M	0.90	**0.83**	**0.65**

Table 3: Model features and resulting classifier accuracies for each of the tested BERT variants.

results in a slight performance improvement for the 1:2 proportion on the ARC and MMLU-Pro datasets (Table 2) when compared to the equal proportion used in Experiment 1. Given that the ARC has a lower proportion of [None] samples than the RFQ, this result does not seem to depend merely on an increased proportion of [None] Class responses in these two datasets (see Table 1). However, when comparing to the results of Experiment 1, even the improved performances were still at best equal to the rule-based classifier's scores.

6.2.3 BERT-variant Model Comparison

Setup. To investigate the impact that the choice (and size) of pre-trained model has on the classifier design, we compare four different BERT model variants. Specifically, we consider the ALBERT (Lan et al., 2020) base model (12M), DistilBERT (Sanh et al., 2019) base cased (66M), DistilRoBERTa base (83M) and RoBERTa (Liu et al., 2019) base (125M), which we chose to cover different size variants of similar encoder models. As in the previous experiments, all classifier variants are tested on all three datasets.

Results. From the results shown in Table 3, we find the best performing models to be the DistilRoBERTa-based classifier and the RoBERTa-based classifier that we used in Experiment 1. Notably, none of the parameter variations for the classifier leads to higher performances than the rule-based classifier. Considering runtimes, we find negligible differences in training times and inference times between the models (see Appendix Table 9).

7 Discussion and Outlook

Our results for closed-ended questions stand in contrast to the findings of Gweon and Schonlau (2023) for classifying LLM responses to open-ended questions, whose fine-tuned neural models outperformed a training-free alternative. While this finding is, to some degree, expected as transformer models should cope better with the semantic variation that is higher in free text responses than in closed-ended questions, we find it particularly interesting that the rule-based classifier has a higher performance even on datasets with a high answer-option complexity, for which one would expect models with better semantic modeling to perform better. Given the already high performance of this simple rule-based classifier originally intended as a baseline, we did not explore more complex rule-based approaches, which might yield even better results. Even for the best fine-tuned model (DistilRoBERTa), its best classification performance on the RFQ data (see Table 3) merely matches the accuracy of the rule-based model. Given the large amount of used training data for fine-tuning in our experiments ($> 35,000$ training samples), we take this as an indication that the underperformance of model-based classifiers cannot solely be attributed to a lack of exposure to suitable training data and that there are fundamental limitations in their capability of identifying answer options patterns in the generated LLM responses.

Given the considerably higher computational costs associated with fine-tuning a model-based classifier, the rule-based classifier appears to be the overall better option in our evaluation. The RoBERTa-based model achieved performance metrics closest to the rule-based method while demonstrating lower training times and comparable classification times to other, even smaller, BERT-based models (see Appendix Table 9). This supports the findings by Cunha et al. (2025) and Gweon and Schonlau (2023), who have also concluded that RoBERTa offers the best cost-performance effectiveness among BERT-based models. However, it remains more resource-intensive, even in its distilled form, than the rule-based alternative. Recently, other works by Cunha et al. (2025), Gweon and Schonlau (2023), and Vajjala and Shimangaud (2025) explored a variety of model-based classification approaches (including fine-tuning BERT models) and showed varying performances and notably no universally top-performing solution, with traditional methods achieving competitive or superior results in several applications than LLM-based approaches.

Finally, in the context of answer classification, distinguishing responses into two categories – Answer Option Class and [None] Class – appears to be a crucial factor in selecting a classifier for categorizing LLM-generated answers. While elegant

in design and flexibility, the entropy-based models struggled to classify [None] Class responses in both the ARC and MMLU-Pro datasets. In contrast, NEM showed a substantially higher performance to these classifier designs, especially in the MMLU-Pro dataset, which contains the highest number of [None] Class answers (see Figure 2). Since the proportion of [None] Class answers may vary significantly depending on the answering model (see Appendix Table 10), this insight may be of crucial importance when selecting a suitable model for classifying the responses of a specific LLM in practice.

In summary, choosing the best tool for answer classification should take into consideration several factors beyond performance metrics, such as the specific survey instrument, the available compute resources, and implementation times. Considering these factors, neural models are a reliable and versatile solution, especially when handling complex outputs such as answers to open-ended questions and noisy responses from lower-performance models, but our findings indicate that in the standard context of closed-ended questionnaires, rule-based methods are a valid and less resource-intensive alternative. Training data and code for our experiments are available on Github[1].

Limitations

While we were careful in the design and execution of our experiments and the selection and handling of the data, we see caveats in the application of our findings – in particular due to the possibility of bias during data annotation.

Impact of Annotation. During the manual inspection of generative LLM responses, we noticed that the measured performances of classifier designs are likely dependent on design decisions in the annotation process. For example, annotating an LLM response of `(3)` for an answer option `3. sometimes` is defensible as both a match and a mismatch, depending on interpretation. While we have no reason to suspect that the relative performance of neural classifiers would necessarily change as a result of a different annotation scheme, their overall performance likely would. In comparison to the rule-based classifier, however, the performance of neural models might improve the more lenient annotators are in accepting semantically similar LLM responses (for details on our annotations, see Appendix A.5).

Impact of LLM size. Our results also reveal that, unsurprisingly, the performance of LLMs in answering questionnaires accurately tends to depend on the size of the model, with smaller LLMs struggling more to generate valid responses. This, of course, also impacts the performance of classifiers – and in particular the rule-based classifier – who must cope with interpreting these noisy outputs.

Data Contamination. Given the prevalence of some of our data sources (in particular the RFQ), there is a risk that the transformer models we used were pre-trained on some of this data. However, as one would expect data contamination to increase the performance in comparison to a rule-based classifier, not decrease it, we do not consider this to be an issue in the interpretation of our findings.

Acknowledgments

We would like to thank Luka Galić for his contribution to the rule-based model's heuristics.

AI Statement

Language model-based AI tools (ChatGPT) were used as coding assistants in the implementation and as writing assistants in drafting parts of the manuscript. The final version of the manuscript was written without the aid of AI.

References

Gati V. Aher, Rosa I. Arriaga, and Adam Tauman Kalai. 2023. Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning, ICML*, pages 337–371.

Lisa P Argyle, Ethan C Busby, Nancy Fulda, Joshua R Gubler, Christopher Rytting, and David Wingate. 2023. Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351.

Marcel Binz and Eric Schulz. 2023. Using cognitive psychology to understand GPT-3. *Proceedings of the National Academy of Sciences*, 120(6):e2218523120.

Bojana Bodroža, Bojana M Dinić, and Ljubiša Bojić. 2024. Personality testing of large language models: limited temporal stability, but highlighted prosociality. *Royal Society Open Science*, 11(10):240180.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind

[1] https://github.com/astrlg/classifiers_for_mcq_llmresponses

Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

Washington Cunha, Leonardo Rocha, and Marcos André Gonçalves. 2025. A thorough benchmark of automatic text classification: From traditional approaches to large language models. *CoRR*, abs/2504.01930.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, and et al. 2024. The Llama 3 herd of models. *CoRR*, abs/2407.21783.

John Fields, Kevin Chovanec, and Praveen Madiraju. 2024. A survey of text classification with transformers: How wide? How large? How long? How accurate? How expensive? How safe? *IEEE Access*, 12:6518–6531.

Andrea Gasparetto, Matteo Marcuzzo, Alessandro Zangari, and Andrea Albarelli. 2022. A survey on text classification algorithms: From text to predictions. *Inf.*, 13(2):83.

Ahmad Ghazal, Tilmann Rabl, Minqing Hu, Francois Raab, Meikel Poess, Alain Crolotte, and Hans-Arno Jacobsen. 2013. BigBench: Towards an industry standard benchmark for big data analytics. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, page 1197–1208, New York, NY, USA. Association for Computing Machinery.

Hyukjun Gweon and Matthias Schonlau. 2023. Automated classification for open-ended questions with BERT. *Journal of Survey Statistics and Methodology*, 12(2):493–504.

Thilo Hagendorff, Ishita Dasgupta, Marcel Binz, Stephanie C. Y. Chan, Andrew Lampinen, Jane X. Wang, Zeynep Akata, and Eric Schulz. 2024. Machine psychology. *Preprint*, arXiv:2303.13988.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300.

E. Tory Higgins, Ronald S. Friedman, Robert E. Harlow, Lorraine Chen Idson, Ozlem N. Ayduk, and Amy Taylor. 2001. Achievement orientations from subjective histories of success: Promotion pride versus prevention pride. *European Journal of Social Psychology*, 31(1):3–23.

John J Horton. 2023. Large language models as simulated economic agents: What can we learn from homo silicus? Working Paper 31122, National Bureau of Economic Research.

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, and 7 others. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.

Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang. 2021. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multim. Tools Appl.*, 80(8):11765–11788.

Arina Kostina, Marios D. Dikaiakos, Dimosthenis Stefanidis, and George Pallis. 2025. Large language models for text classification: Case study and comprehensive review. *CoRR*, abs/2501.08457.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025. From generation to judgment: Opportunities and challenges of LLM-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2022. A survey on text classification: From traditional to deep learning. *ACM Trans. Intell. Syst. Technol.*, 13(2).

Xingxuan Li, Yutong Li, Lin Qiu, Shafiq Joty, and Lidong Bing. 2024. Evaluating psychological safety of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1826–1843, Miami, Florida, USA. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Xinyi Lu and Xu Wang. 2024. Generative students: Using LLM-simulated student profiles to support question item evaluation. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, L@S '24, page 16–27, New York, NY, USA. Association for Computing Machinery.

Yang Lu, Jordan Yu, and Shou-Hsuan Stephen Huang. 2023. Illuminating the black box: A psychometric investigation into the multifaceted nature of large language models. *Preprint*, arXiv:2312.14202.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2022. Deep learning-based text classification: A comprehensive review. *ACM Comput. Surv.*, 54(3):62:1–62:40.

Marilù Miotto, Nicola Rossberg, and Bennett Kleinberg. 2022. Who is GPT-3? An exploration of personality, values and demographics. In *Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS)*.

Max Pellert, Clemens M Lechner, Claudia Wagner, Beatrice Rammstedt, and Markus Strohmaier. 2024. AI psychometrics: Assessing the psychological profiles of large language models through psychometric inventories. *Perspectives on Psychological Science*, 19(5):808–826.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Julian Schelb, Orr Borin, David Garcia, and Andreas Spitz. 2025. R.u.psycho? Robust unified psychometric testing of language models. *Preprint*, arXiv:2503.10229.

Matthias Schonlau, Julia Weiß, and Jan Marquardt. 2023. Multi-label classification of open-ended questions with BERT. *CoRR*, abs/2304.02945.

Gregory Serapio-García, Mustafa Safdari, Clément Crepy, Luning Sun, Stephen Fitz, Peter Romero, Marwa Abdulhai, Aleksandra Faust, and Maja Matarić. 2025. A psychometric framework for evaluating and shaping personality traits in large language models. *Nature Machine Intelligence*, 7(12):1954–1968.

S. S. Stevens. 1946. On the theory of scales of measurement. *Science*, 103(2684):677–680.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005, Singapore. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, and et al. 2023. Zephyr: Direct distillation of LM alignment. *CoRR*, abs/2310.16944.

Sowmya Vajjala and Shwetali Shimangaud. 2025. Text classification in the LLM era - Where do we stand? *CoRR*, abs/2502.11830.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. 2024a. "My Answer is C": First-token probabilities do not match text answers in instruction-tuned language models. In *Findings of the Association for Computational Linguistics, ACL*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024b. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Hanzi Xu, Renze Lou, Jiangshu Du, Vahid Mahzoon, Elmira Talebianaraki, Zhuoan Zhou, Elizabeth Garrison, Slobodan Vucetic, and Wenpeng Yin. 2024. LLMs' classification performance is overclaimed. *CoRR*, abs/2406.16203.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, and et al. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Yazhou Zhang, Mengyao Wang, Qiuchi Li, Prayag Tiwari, and Jing Qin. 2025. Pushing the limit of LLM capacity for text classification. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 1524–1528, New York, NY, USA. Association for Computing Machinery.

A Classifier Data

A.1 Answer Classifier Data

To systematize answer types that the tested classifier versions should handle, we choose the traditional classification of Stevens (1946), due to its exhaustiveness and simplicity. Question types are simply classified on the answer data type (nominal, ordinal, interval, ratio). Nominal data comprises responses that fall into discrete categories (e.g., True or False questions, or multiple-choice questions), where answer options may also be numbered, but these numbers do not imply any order, e.g., *Question: "Which of these supermarkets in your opinion sells the best-quality fresh vegetable?", Answer Options: "1. Asda, 2. Morrisons, 3. Safeway, 4. Sainsbury's, 5. Somerfield, 6. Tesco"*. Ordinal data instead requires respondents to arrange nominal categories based on a specific criterion outlined in the question such as ranking scales, e.g., *Question: "How confident do you feel today?", Answer Options: "Very Confident, Somewhat Confident, Not Confident"*. Interval scales feature answer items arranged on a scale with an arbitrary zero point, where the distance between each point is numerically equal, these are usually rating scales such as Likert scales, Stapel scales, and Semantic Differential Scales, e.g., *Question: "Please rate our product between 1 to 5 on how much you are satisfied.", Answer Options: "1 star, 2 stars ,3 stars, 4 stars, 5 stars"*. Finally, a ratio scale is a specific type of interval scale in which there is a meaningful zero point, e.g., *Question: "Of the last 10 cans of baked beans that you bought, how many were Heinz?", Answer Option: "None, 1, 2, 3, 4, 5"*. Further subdivisions can be applied to answer types: the number of possible answer options (2 or more), scales can be balanced or unbalanced, unipolar or bipolar, partly labeled or completely labeled, with or without a middle point, etc. We do not consider such fine-grained distinctions to be meaningful for response classification in our evaluation context, and therefore do not intentionally include them in our benchmark datasets.

A.2 Training Dataset Generation: Positive Samples

A.2.1 Answer Option Class

Response Template Generation. To generate training data for the Answer Option Class, we first use the same dictionary of 67 handcrafted templates from Schelb et al. (2025). These templates are created with the aim of resembling outputs from question-answering models, e.g., `The evidence strongly supports ⟨answer option⟩`, where `⟨answer option⟩` is then filled in with an answer option, e.g., `The evidence strongly supports 3. sometimes`. Each of the templates is combined with all possible answer options. The answer options consist of the answer items from the Regulatory Focus Questionnaire (RFQ) (Higgins et al., 2001). There are four different answer option sets in the RFQ, these are: `[1. never or seldom, 2., 3. sometimes, 4., 5. very often]`; `[1. certainly false, 2., 3., 4., 5. certainly true]`; `[1. never or seldom, 2., 3. sometimes, 4., 5. always]`; `[1. never or seldom, 2., 3. sometimes, 4., 5. many times]`. The complete RFQ is reported below in the relevant Appendix Section A.5. The RFQ answer options are modified so as to include all basic answer formats, i.e., only numeric, only non-numeric, or mixed. Therefore the answer options found in each template can consist of only the answer class (e.g. `1.`), or only the answer label (e.g. `never or seldom`), or both (e.g. `1. never or seldom` or `never or seldom 1.`). Each template is filled with each of the three variations for each of the answer options. The mixed-type answers are split across templates between variation 1 (`1. never or seldom`) and variation 2 (`never or seldom 1.`). In total 2145 filled templates, uniformly sampled across answer option variations, are generated.

Paraphrased Templates. To increase the diversity of the Answer Option Class, we generate additional examples by instructing Llama 3.1 70B (Dubey et al., 2024) to paraphrase the filled-in templates. Only the Answer Option Class templates require augmentation and therefore they are the only paraphrased templates. This is the initial set of all positive training samples for the Answer Option Class. The same paraphrasing model, prompt and instructions as in Schelb et al. (2025) are used. The paraphrasing model is instructed to generate multiple distinct paraphrases of a given statement from the original templates, we randomly select 20 strategies from a set of 61 handcrafted predefined instructions for generating paraphrases. The resulting generated sentences are separated by newlines and filtered for empty values to create several paraphrased versions of the original statement. Any paraphrased sample with a cosine similarity below

the 25th percentile, between generated paraphrase and reference template, is then discarded. The cosine similarity score is computed using Sentence-BERT embeddings (Reimers and Gurevych, 2019). The remaining valid paraphrases amount to 26500 samples.

The filled-in handcrafted templates and the paraphrased templates are then combined into a single dataset. Samples from this dataset are then filtered for duplicates and sampled uniformly across answer option variations (i.e., numeric, non-numeric, mixed). Each answer option (e.g., `1. never or seldom`) ended up having 792 positive samples overall, of which 1/3 (264 samples) represent the answer option in its numeric-only variation (e.g., `1.`), 1/3 represent the answer option in its "non-numeric" variation (e.g., `never or seldom`), and 1/3 in its mixed variation (e.g., `1. never or seldom`). Where the answer options are only numerical (i.e, answer options `2.` and `4.`), we sample 792 positive samples for the numeric-only variation. In total, considering both templates and paraphrases, there are 10296 positive samples for the Answer Option Class.

A.2.2 [None] Class

[None] Class Samples. For the [None] Class, a distinction is made between two types of [None] Class answers, these are *inconclusive* and *not present* answers. Inconclusive answers refer to responses which are ambiguous, i.e., they contain answer options without a clearly identifiable choice, e.g. `The correct option would be 5. very often or 2.` where the valid answer options are `1. never or seldom, 2., 3. sometimes, 4., 5. very often`. Not present answers are instead responses in which no answer option is present and the response is irrelevant with regard to the question, e.g. `As I reflect on my life, I would say that I feel like I have indeed made progress toward being successful in my life.`, where the valid answer options are `1. never or seldom, 2., 3. sometimes, 4., 5. very often`.

Not Present [None] Samples. Not present [None] Class answers are not generated from handcrafted templates, but from a combination of two subsets. The first subset consists of discarded paraphrased templates from the Answer Option Class. These are paraphrases with a cosine similarity between paraphrased response and answer option, calculated using Sentence-BERT embeddings, below the 10th percentile. The second subset consists of "real-world" irrelevant outputs from three models (Qwen2.5 3B, Llama 3.1 8B, Llama 3.1 70B) prompted to answer the RFQ questions. The prompt given to the models is the following: "*Question: <question item>*", where question item is replaced by each of the RFQ questions. The models are prompted to respond without being given the list of valid answer options for each question. We then randomly sample 264 answers from the two combined subsets, to match the amount of positive samples for each answer option variation of the Answer Option Class.

Inconclusive [None] Samples. Inconclusive templates can be of two types: mismatched or multiple-answer. In the mismatched case (e.g., `1. certainly true`) an answer class number (e.g., `1.`) is attributed to the wrong answer label (e.g. `certainly true`). In the multiple-answer case instead any two answer options are present in the template (e.g., `very often 5.` or `sometimes 3.`). The same 67 handcrafted templates from the base dataset are adapted and expanded to include two answer options in the mismatched or multiple-answer version. Each template is then completed by sampling a random combination of two different answer options, we repeat the sampling 10 times. After filtering for empty values and duplicates, and after uniformly sampling across answer option variations, we then populate the final inconclusive dataset with 264 samples for the mismatched templates and 264 samples for the multiple-answer templates. As for the Answer Option Class, also the inconclusive answer options have 264 positive samples for each variation.

The [None] Class Templates amount in total to 792 samples, i.e., equal to the number of samples for each answer option of the Answer Option Class (with 264 samples for each of the three variations of an answer option). All positive samples (Answer Option Class and [None] Class combined) amount to 11088 in total.

A.3 Training Dataset Generation: Negative Samples

Negative samples are then generated for the Answer Option Class and the [None] Class *not present* samples by randomly assigning incorrect answer options to each previously response sample, and labeling them as non-corresponding. An example

of a negative sample for the Answer Option Class is the following: `As an AI language model, I cannot help with that` (this is a [None] Class sample), paired with the answer option `3. sometimes`, and labeled `0`. An example of a negative sample for the [None] Class *not present* samples is instead: sample `The evidence strongly supports 3.sometimes` (this is an Answer Option Class sample), paired with the answer option `[None]`, and labeled `0`. Negative samples for [None] Class *inconclusive* samples instead are not created by randomly sampling answer options from the Answer Option Class. Instead, the answer option to pair with the inconclusive sample, is randomly sampled from one of the two answer options present in the inconclusive response sample. For example if the inconclusive sample is `My choice is 3. or 4.`, this response sample is paired with either answer option `3.` or `4.`, and labeled `0`. For each positive sample, 3 negative samples are generated, for a total of 33264 negative samples (Answer Option Class and [None] Class combined).

A.4 Final Training Dataset Generation

The final training data is created by merging together both positive and negative samples from the Answer Option Class and the [None] Class, for a total of 44352 samples. This dataset is then split with an 80/20 ratio into a train dataset (35481 samples) and a validation dataset (8871 samples). This is the resulting training dataset used for the NEM classifier. The training datasets for the other three model-based classifier variations are generated similarly by simple exclusion of a single sample subset depending on the model design. For the EM-O the excluded subset is the whole [None] Class, for the EM-1 model this is the whole set of negative samples for the [None] Class, and for the EM-2 model this is the subset of positive samples for the [None] Class.

A.5 Test Dataset Generation

A.5.1 Dataset 1: RFQ

The same annotated dataset from Schelb et al. (2025) is repurposed and expanded. This dataset consists in total of 2,750 responses to the Regulatory Focus Questionnaire (RFQ) (Higgins et al., 2001) for each of three prompt variants (reported below), 25 different personas (randomly sampled from the list of names published by Aher et al., 2023, each in combination with the title Ms. or Mr.) and seven different models: Qwen 2.5 0.5B and 7B and 32B and 72B (Yang et al., 2024), Llama 3.1 8B and 70B (Dubey et al., 2024), Zephyr 7B (Tunstall et al., 2023). This results in a total of 41,250 responses. Of these, 500 responses were randomly sampled and annotated independently by two annotators (one with a psychology/data science background and one with a computer science background). This dataset was then expanded by randomly sampling an additional 500 responses from the initial response dataset. An additional answering model was then added (Gemini 3 Pro, Team et al., 2025) and prompted (with the same three prompt variants) to answer 132 additional question items. The additional two subsets were annotated separately by two different annotators than the first 500 answers (one with a psychology/data science background and one with a linguistics background). Responses with disagreeing annotations were excluded. The final dataset consists of 1062 annotated answers.

Questionnaire instruction: *This set of questions asks you HOW FREQUENTLY specific events actually occur or have occurred in your life. Please indicate your answer to the question by selecting the appropriate number.*

1. Question: *Compared to most people, are you typically unable to get what you want out of life?*
Answer Options: *1. never or seldom, 2., 3. sometimes, 4., 5. very often*

2. Question: *Growing up, would you ever "cross the line" by doing things that your parents would not tolerate?*
Answer Options: *1. never or seldom, 2., 3. sometimes, 4., 5. very often*

3. Question: *How often have you accomplished things that got you "psyched" to work even harder?*
Answer Options: *1. never or seldom, 2., 3. sometimes, 4., 5. many times*

4. Question: *Did you get on your parents' nerves often when you were growing up?*
Answer Options: *1. never or seldom, 2., 3. sometimes, 4., 5. very often*

5. Question: *How often did you obey rules and regulations that were established by your parents?*
Answer Options: *1. never or seldom, 2., 3. sometimes, 4., 5. always*

6. Question: *Growing up, did you ever act in ways that your parents thought were objectionable?*
Answer Options: *1. never or seldom, 2., 3. sometimes, 4., 5. very often*

7. Question: *Do you often do well at different things that you try?*

Answer Options: *1. never or seldom, 2., 3. sometimes, 4., 5. very often*
8. Question: *Not being careful enough has gotten me into trouble at times.*
Answer Options: *1. never or seldom, 2., 3. sometimes, 4., 5. very often*
9. Question: *When it comes to achieving things that are important to me, I find that I don't perform as well as I ideally would like to do.*
Answer Options: *1. never or seldom, 2., 3. sometimes, 4., 5. very often*
10. Question: *I feel like I have made progress toward being successful in my life.*
Answer Options: *1. certainly false, 2., 3., 4., 5. certainly true*
11. Question: *I have found very few hobbies or activities in my life that capture my interest or motivate me to put effort into them.*
Answer Options: *1. certainly false, 2., 3., 4., 5. certainly true*

Prompt 1.
Instructions: Choose from the list of answer options to answer the question. Answer the question using only the provided answer options. If none of the options are correct, choose the option that is closest to being correct.
Question: <instruction item>
Answer Options: <answer options>
Answer:

Prompt 2.
Objective: Act like you are ⟨persona⟩, a survey participant answering a questionnaire. ⟨questionnaire instruction⟩
Instructions: Choose from the list of answer options to answer the question. Answer the question using only the provided answer options. If none of the options are correct, choose the option that is closest to being correct.
Question: <instruction item>
Answer Options: <answer options>
Answer:

Prompt 3.
Objective: Act like you are ⟨persona⟩, a survey participant answering a questionnaire. ⟨questionnaire instruction⟩
Instructions: Choose from the list of answer options to answer the question. Answer the question using only the provided answer options. If none of the options are correct, choose the option that is closest to being correct.
Output format: The solution must be provided in this format: {"answer": "answer option"}.
Question: <instruction item>
Answer Options: <answer options>
Answer:

A.5.2 Dataset 2: ARC

The second dataset consists of the AI2 Reasoning Challenge (ARC) (Clark et al., 2018). The dataset consists of a corpus of grade-school science questions each with four possible answer options to choose from. Out of 7,787 available questions, we randomly sampled 500 questions, of which 250 questions come from the Easy subset and 250 questions come from the Challenge subset. Each answer-option set was additionally randomly assigned either a numerical (i.e., 1., 2., 3., 4.), alphabetical (i.e., A., B., C., D.) or no answer label. Then each of five different answering models was prompted to answer a randomly sampled subset of 100 questions, out of the 500 questions previously sampled. The prompted models were: Qwen 2.5 72B, Llama 3.1 70B, Zephyr 7B, Gemini 3 Pro and Gemma 2 9B (Team et al., 2024). Each model was prompted to respond using the second prompt used in the RFQ adapted to this dataset. The prompt was chosen as to be a prompt of medium complexity. The same annotators who annotated the expanded RFQ dataset annotated the ARC dataset as well. Items annotated differently were discarded. The final dataset consists of 481 annotated answers.

Prompt.
Instructions: Choose from the list of answer options to answer the question. Answer the question using only the provided answer options. If none of the options are correct, choose the option that is closest to being correct.
Question: <instruction item>
Answer Options: <answer options>
Answer:

A.5.3 Dataset 3: MMLU-Pro

The questions included in the dataset consist of items from the Measuring Massive Multitask Language Understanding Pro (MMLU-Pro) task dataset (Wang et al., 2024b), which expands the MMLU task dataset (Hendrycks et al., 2020) to include more questions and number of possible answer options. This dataset comprises items with

four to ten answer options of nominal, interval and ratio type. In addition, we modified the MMLU-Pro dataset to have equal subsets of questions differing by number of answer options. This was achieved by reducing the number of answer options for a portion of the 10- and 9-answer option subset (the most numerous question subgroup in the original dataset). Randomly sampled questions were selected from this subset, and then the correct answer option was retained with the addition of other randomly sampled answer options from the question's answer list. For example, to populate the subset of 3-answer option questions, we kept the correct answer to the question and randomly sampled two other items from the answer option list of the question item. To further increase the variability of the answers, the answer options were additionally randomly assigned numerical, alphabetical or no answer labels (as done for the ARC dataset). The original test dataset consists of 12.032 questions divided into 14 topic categories (e.g. animals, movies, sports, etc.), from which an equal number of answers is sampled based on number of answer options. The resulting sampled subset consists of 500 question items. This benchmark dataset was created by asking five different models to each respond to one fifth of the items from the sampled set of questions. The prompted models were: Qwen 2.5 72B, Llama 3.1 70B, Gemma 2 9B and Mixtral 8x7B (Jiang et al., 2024). In addition to the initial 500 items, 126 more questions were sampled (again uniformly across answer option number and topic) to prompt Gemini 3 Pro. The models were chosen based on relevance in the field, differing sizes and reported performances on the MMLU-Pro task. The answers from all models were then merged and, after filtering out empty answers and duplicate question-answer pairs, each response was labeled with the most likely answer option. The same annotators from the RFQ expanded dataset and the ARC dataset annotated this dataset as well. Items with differing annotations were discarded. The final dataset consists of 498 annotated answers. The prompt for the MMLU-Pro dataset was chosen to have the less complex format possible, while still retaining a common part to the other datasets' prompts.

Prompt.
Question: <instruction item>
Answer Options: <answer options>
Answer:

A.5.4 Dataset Annotations

The RFQ dataset was annotated by four different annotators. 500 answers were annotated independently by two annotators (one with a computer science background and one with a psychology/data science background). The remaining 626 answers were annotated by a different pair of annotators, again independently from each other (one with a linguistics background and one with a psychology/data science background). Where annotations did not match, the response samples were discarded. For the RFQ dataset 10 out of 1078 answers were discarded. For the ARC 19 answers out of 500 were discarded. For the MMLU-Pro 126 out of 600 answers were discarded. The increasingly higher number of disagreeing annotations correlates with the increasing difficulty of the question sets, resulting in progressively noisier LLM outputs that were challenging to classify also for the annotators. Annotations for all three datasets followed the same annotation guidelines (reported below).

Answer Option Class. If one and only one answer option was present in the response sample and unambiguously identifiable, the response sample was labeled with that answer option, for example:

> **Answer Options**: `1. never or seldom, 2., 3. sometimes, 4., 5. very often, [None]`
> **Response Sample**: `I would select answer option 1. never or seldom.`
> **Annotation**: `1. never or seldom`

When partial answer options were present, they were matched to an answer option only if the answer option was uniquely and unambiguously identifiable, for example:

> **Answer Options**: `1. never or seldom, 2., 3. sometimes, 4., 5. very often, [None]`
> **Response Sample**: `seldom`
> **Annotation**: `1. never or seldom`

[None] Class. Response samples labeled as [None] were of five possible types:
i) No identifiable answer option was present in the response sample, for example:

> **Answer Options**: `1. never or seldom, 2., 3. sometimes, 4., 5.`

very often, [None]
Response Sample: As I reflect on my life, I would say that I feel like I have indeed made progress toward being successful in my life.
Annotation: [None]

ii) Multiple answer options were present in the response sample, with no clear choice, for example:

Answer Options: 1. never or seldom, 2., 3. sometimes, 4., 5. very often, [None]
Response Sample: The correct option would be 5. very often or 2.
Annotation: [None]

iii) An answer option with the incorrect numerical or alphabetical label was present in the response, for example:

Answer Options: 1. never or seldom, 2., 3. sometimes, 4., 5. very often, [None]
Response Sample: 4. very often
Annotation: [None]

iv) Partial answer options were present in the response sample but were not sufficient to uniquely and unambiguously match them to a single answer option, for example:

Answer Options: 1. never or seldom, 2., 3. sometimes, 4., 5. very often, [None]
Response Sample: often
Annotation: [None].

Here "often" could be matched to both answer option 4. and answer option 5. very often
v) A paraphrased answer option was present in the response without numerical, alphabetical or other labeling for unambiguous answer option identification, for example:

Answer Options: 1. never or seldom, 2., 3. sometimes, 4., 5. very often, [None]
Response Sample: Ms. Nez: Compared to most people, I would say that I am able to get what I want out of life more often than not. While there may have been occasional setbacks or challenges, I would say that I am rarely, if ever, unable to achieve my goals,
Annotation: [None]

Models	Positive Class		[None]Class	
	Positive samples	Negative samples	Positive samples	Negative samples
NEM	✓	✓	✓	✓
EM-1	✓	✓	✓	
EM-2	✓	✓	✓	
EM-O	✓	✓		
RbC				

Table 4: Training data comparison of all tested classifier versions.

B Models

In Experiment 1 we compare four different model-based classifiers. The four models differ in training data composition (see Table 4) and in the method used for the [None] Class detection. An explanation of each model is provided below.

B.1 Rule-based Classifier (RbC)

This model is based on token overlap to determine which answer option exhibits the highest lexical similarity to the response generated by an LLM answering a question item. Initially, each answer option is divided into two parts: a label part and an answer class part (for example, 5. and always). The frequency of token overlap within each part of the response is counted. The answer option that accumulates the highest total overlap score is identified as the best choice. If two answer options share the same score (*inconclusive*) or there is no token overlap (*not present*) then the outcome is classified as [None]. Additional heuristics are added to handle both numerical and non-numerical labeling, text normalization to lower-case and exclusion of only partially-overlapping answer options.

B.2 Entropy Models (EM-1, EM-2, EM-O)

The entropy-based models (EM-1, EM-2, EM-O) closely resemble the model-based judge from Schelb et al. (2025), and differ mostly in the training data composition. The entropy-based models consist of a BERT-based (RoBERTa, Liu et al., 2019) fine-tuned classifier, which assigns probabilities to input pairs consisting of i) an LLM-generated response to a question item and ii) an

Model	RFQ	ARC	MMLU-Pro
EM-1	0.54	1.0	1.0
EM-2	0.81	0.98	1.0
EM-O	0.49	1.0	1.0

Table 5: Entropy Thresholds used in Experiment 1 for models EM 1, EM 2 and EM-O on each benchmark dataset.

answer option to the question item. After computing the probability for each match (i.e., for each response-answer option pair), this classifier computes the entropy of the probability distribution of the input-pair values. It then matches responses to the option [None], only when the entropy value is above an experimentally-found threshold. The entropy (H) is normalized to account for differing numbers of answer options and is calculated as

$$H(X) = \frac{H_{\text{max}} - H(X)}{H_{\text{max}} - H_{\text{min}}}$$

where

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log(p(x_i)).$$

The three entropy-based models all share the design described above, but differ in the representation of the [None] Class in their training dataset (see Table 4). The Entropy-based Model Original (EM-O) corresponds to the model-based judge developed by Schelb et al., 2025). For this classifier version, the [None] Class is completely absent from the training data. For the Entropy-based Model version 1 (EM-1), the training dataset does also not include negative samples for the [None] Class, however [None] Class response samples are present in this case among the negative samples for the Answer Option Class. For the Entropy-based Model version 2 (EM-2) instead, the training dataset does also include negative samples for the [None] Class. The three entropy-based classifier versions therefore differ in the degree to which they include the [None] Class in their training datasets. All other features remain unchanged between these entropy-based classifier variations.

Entropy Threshold. For each of the models using an entropy-threshold to classify the [None] Class (i.e., EM 1, EM 2 and EM-O), the entropy threshold to be used was determined by testing threshold values between 0 and 1 (given that the entropy is normalized) in steps of 0.01. Each of the threshold values was used for classification on each of the three test datasets and performance was then calculated. For each model evaluated in each of the three datasets the entropy-threshold value yielding the best classification performance was selected for Experiment 1. For each model and test dataset a different optimal entropy threshold was found. The final entropy thresholds for each of the models are shown in Table 5.

B.3 No-Entropy Model (NEM)

The No-Entropy Model (NEM) is also a BERT-based (RoBERTa, Liu et al., 2019) classifier, which is fine-tuned to evaluate each response by comparing it against all possible answer options. The input pairs in this case do not only include the answer options predefined in the questionnaire, but there is an additional match to evaluate, consisting of the response and the [None] Class option. This model therefore evaluates the match probability between response and answer option by adding among the possible answer options the [None] option representing the [None] Class (see Figure 1). The No-Entropy Model was the best performing model-based classifier in Experiment 1 and therefore was selected as the classifier version to be tested in Experiment 2.

Model	Training Data Size	Training Time (min)	Classification Time (min)			Classification Accuracy		
			RFQ	ARC	MMLU -Pro	RFQ	ARC	MMLU -Pro
NEM	35481	15.79	1.04	0.39	0.57	0.90	0.83	0.65
EM-1	32946	8.17	2.18	0.82	2.07	0.88	0.87	0.62
EM-2	34847	8.43	2.11	0.97	1.03	0.89	0.86	0.60
EM-O	32946	8.20	1.10	0.45	1.03	0.88	0.88	0.61
RbC	0	0	0.010	0.01	0.01	**0.96**	**0.90**	**0.74**

Table 6: Training datasets and classification details for each classifier version tested in Experiment 1, including runtimes on an Nvidia A40 GPU.

Model	Accuracy			Precision		
	RFQ	ARC	MMLU-Pro	RFQ	ARC	MMLU-Pro
NEM	0.90	0.83	0.65	0.91	0.83	**0.73**
EM-1	0.88	0.87	0.62	0.88	0.87	0.62
EM-2	0.89	0.86	0.60	0.88	0.86	0.60
EM-O	0.88	0.88	0.61	0.88	0.88	0.61
RbC	**0.96**	**0.90**	**0.74**	**0.96**	**0.88**	0.72
Model	**Recall**			**F1**		
	RFQ	**ARC**	**MMLU-Pro**	**RFQ**	**ARC**	**MMLU-Pro**
NEM	0.90	0.83	0.65	0.88	0.83	0.68
EM-1	0.92	0.87	0.62	0.89	0.87	0.62
EM-2	0.89	0.86	0.60	0.87	0.86	0.60
EM-O	0.88	0.88	0.64	0.87	0.88	0.61
RbC	**0.96**	**0.90**	**0.74**	**0.96**	**0.88**	**0.73**

Table 7: Performance metrics for each tested classifier version in Experiment 1.

Negative Samples Ratio	Training Data Size (# samples)	RFQ	ARC	MMLU-Pro
1:1	17740	0.90	0.80	0.67
1:2	26610	0.90	0.80	0.60
1:3	35481	0.90	**0.83**	**0.65**
1:4	44351	0.81	0.67	0.61
1:5	53222	0.90	0.72	0.53
1:6	62092	0.90	0.72	0.59
1:7	70962	**0.92**	0.79	0.66
1:8	79833	0.91	0.68	0.54
1:9	88703	0.90	0.80	0.64
1:10	97574	0.85	0.62	0.56

Table 8: Training datasets and classification details across ten different negative sample ratios tested in Experiment 2 Negative Sample Ratio.

Model	Size	Training Time (min)	Classification Time (min)			Classification Accuracy		
			RFQ	ARC	MMLU-Pro	RFQ	ARC	MMLU-Pro
ALBERT	12M	19.62	0.92	0.34	0.51	0.78	0.51	0.43
DistilBERT	66M	13.56	0.56	0.21	0.32	0.87	0.44	0.48
DistilRoBERTa	83M	16.12	0.60	0.23	0.34	**0.96**	0.60	0.59
RoBERTa	125M	15.79	1.04	0.39	0.57	0.90	**0.83**	**0.65**

Table 9: List of BERT-based classifiers and their characteristics compared in Experiment 2 BERT-variant Model Comparison.

Model	Dataset								
	RFQ								
	Answer Length	**Answer Time (min)**	**Model Performance**	**Class Proportions**	**EM-O**	**EM-1**	**EM-2**	**NEM**	**RbC**
Qwen 2.5-0.5B	2.2	n/a	n/a	0.70/0.30	0.72	0.87	0.70	0.70	0.96
Qwen 2.5-7B	3.5	n/a	n/a	1.00/0.00	0.94	0.96	1.00	1.00	0.97
Qwen 2.5-32B	21.7	n/a	n/a	0.98/0.02	0.88	0.76	0.91	0.94	0.98
Qwen 2.5-72B	18.5	n/a	n/a	1.00/0.00	0.92	0.80	0.94	0.98	0.99
Llama 3.1-8B	14.6	n/a	n/a	0.97/0.03	0.96	0.95	0.97	0.97	0.98
Llama 3.1-70B	3.7	n/a	n/a	1.00/0.00	0.91	0.97	1.0	0.99	0.99
Zephyr 7b-beta	40.0	n/a	n/a	0.62/0.38	0.73	0.76	0.71	0.71	0.81
Gemini 3 Pro	6.1	45.5	n/a	1.00/0.00	1.0	0.99	0.99	1.00	1.00
	ARC								
	Answer Length	**Answer Time (min)**	**Model Performance**	**Class Proportions**	**EM-O**	**EM-1**	**EM-2**	**NEM**	**RbC**
Qwen 2.5-72B	26.4	10.47	0.89	0.99/0.01	0.90	0.88	0.92	0.88	0.93
Llama 3.1-70B	29.6	10.80	0.76	0.88/0.12	0.74	0.72	0.75	0.65	0.73
Gemma 2-9b-it	15.6	4.49	0.89	1.00/0.00	0.96	0.92	0.88	0.92	0.95
Zephyr 7b-beta	25.0	3.96	0.73	0.96/0.04	0.80	0.82	0.77	0.72	0.85
Gemini 3 Pro	6.42	10.67	0.91	1.00/0.00	1.0	1.0	0.97	0.97	1.0
	MMLU-Pro								
	Answer Length	**Answer Time (min)**	**Model Performance**	**Class Proportions**	**EM-O**	**EM-1**	**EM-2**	**NEM**	**RbC**
Qwen 2.5-72B	85.7	28.09	0.48	0.76/0.24	0.59	0.58	0.57	0.61	0.70
Llama 3.1-70B	67.2	28.66	0.54	0.92/0.08	0.61	0.69	0.60	0.65	0.71
Mixtral 8x7B	32.9	22.07	0.42	0.87/0.13	0.72	0.77	0.71	0.70	0.83
Gemma 2-9b-it	75.8	17.92	0.44	0.77/0.23	0.63	0.56	0.65	0.67	0.69
Gemini 3 Pro	85.8	28.09	0.49	0.64/0.36	0.55	0.57	0.54	0.63	0.74

Table 10: List of answering models used to generate questionnaire responses to be classified. Answer Length is measured in number of tokens. Model performance refers to the answering model's proportion of correct answers in the task. Class Proportions represent the proportions of responses between the Answer Option Class and the [None] Class. The classification accuracy on the responses from each answering LLM is reported for each classifier version.