Enhancing the Reliability of LLMs-based Systems for Survey Generation through Distributional Drift Detection

Vinicius Monteiro de Lira[1,*,†], Antonio Maiorino[1,†] and Peng Jiang[2,†]

[1]SurveyMonkey, Padua, Italy
[2]SurveyMonkey, San Mateo, California, USA

Abstract

Evaluating Large Language Model (LLM)-based systems is a recurrent challenge in modern machine learning research and development. It is crucial to ensure that any changes made in the production environments will not negatively impact user experience, and clever evaluation techniques are especially important when updated models or prompts create disparities within the system. Since we released the feature to help our customers create surveys with textual prompts in 2023, we have iteratively improved several parts of the system such as the prompts, the LLM models and the system's internal logic. To measure the impact of these changes, we propose a comprehensive framework for assessing surveys generated by LLMs, focusing on data drift analyses based on survey metadata features. By leveraging this approach, we can effectively identify and address potential areas of concern related to model performance, enhancing the reliability and usability of LLM-based systems for survey generation tasks.

Keywords

LLMs, Survey Generation, Reliability, Distribution Drifts

1. Introduction

We are the global leader in survey software, with our flagship platform enabling the collection of over 20 million answers per day across a vast variety of domains. One of the major goals of our service is to help customers create high-quality surveys by leveraging the wealth of research that internal teams have accumulated over the course of many years in the industry. This translates into the continuous development of features aimed at helping customers create effective surveys that allow them to learn what they're interested in by asking the best possible questions to their audience.

One of the latest features released for this purpose is called *Build with AI (BWAI)*. This feature has been released to all the users of the platform near the end of 2023 and leverages Large Language Models (LLMs) to allow users to build high quality surveys through a conversational interface, where users can specify what they want to learn about their audience through a textual description (a *prompt*), which will be used by the system to generate a survey with relevant questions and context.

Since this application involves generating a long text based on concise instructions provided in a short "seed" input text, it can be particularly challenging because of the very nature of the task, which is more akin to "creative writing" than to other Natural Language Processing (NLP) use cases where "correct" and "wrong" labels could typically be identified. In fact, in this use case it is much harder to determine what a "good" or "bad" generation would look like, as there are many possible examples of good surveys in which could be generated starting from the same input prompt.

This paper presents a novel contribution in the form of a comprehensive framework for evaluating surveys generated by LLMs, specifically addressing challenges in survey generation tasks. The framework exploits survey metadata to facilitate data drift analysis, enabling the identification and mitigation of potential issues related to model performance. By systematically analyzing survey metadata and detecting distributional drift, the framework assesses the behavior of LLM-based systems for survey generation.

2. Related Work

The Survey Generation domain that we studied in this work poses its own set of challenges since typically there is not a "correct" or "wrong" survey, but rather the goodness of the model lies in its ability to follow the instructions specified by the user, while also trying to produce interesting ideas for potentially useful survey questions.

This puts our model in an area closer to use cases such as brainstorming and creative writing than to other, more studied areas such as Question Answering, Intent Recognition and Summarization, where some "ground truths" are usually available and can be used to evaluate the level

KiL'24: Workshop on Knowledge-infused Learning co-located with 30th ACM KDD Conference, August 26, 2024, Barcelona, Spain
*Corresponding author.
†These authors contributed equally.
vmonteirodelira@surveymonkey.com (V. M. d. Lira); amaiorino@surveymonkey.com (A. Maiorino); pjiang@surveymonkey.com (P. Jiang)
0000-0002-7580-1756 (V. M. d. Lira)
© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

of quality of the generated text. The lack of ground truth combined with the lack of standardized metrics for open-ended tasks makes evaluation even more difficult in our scenario.

As pointed out in [1] these kinds of use cases are often missing from popular benchmarks such as HELM [2], since most of these tend to focus on verifiable, closed-ended and automated metrics. For reliably evaluating open-ended use cases researchers and practitioners often hire human raters, as for example done by the authors of [1] who hired 10 human raters to evaluate several LLMs on Creative Writing tasks. While a human evaluation is currently still the most effective and reliable way of evaluating such open-ended tasks, human involvement also makes the process much longer and expensive.

Some strategies proposed in the literature to automatically evaluate the quality of open-ended use cases include measuring the degree of "text quality" through metrics such as text readability and diversity. In [3] the authors distinguish between "reference-based metrics", where the output generated by the model is compared to a similar output written by a human, and "reference-free metrics", where the quality of the outputs is measured directly, with some examples of the latter group being n-gram based metrics such as "Lexical Repetition" [4] and "Distinct-3 (D-3) [5]", descriptive statistics such as text length, Self-BLEU (SBL) [6] and BARTScore (BAS) [7]. Nonetheless, the authors also report that these metrics often do not seem to agree with each other, and they complement their assessment with human-based measurements.

The authors of [8] analyze the NLG evaluation landscape from another angle that's becoming more widespread after the advent of big-scale powerful models, which is the LLM-based evaluation. These techniques involve using LLMs themselves as "judges" for generated text and include Scoring, Comparison, Ranking, and Boolean QA among the strategies used to constraint LLMs to output close-ended scores. This direction is exciting because it seems like a promising way to automate evaluation tasks that were previously very hard to automate without models capable of understanding all the nuances in the generated examples, but it also comes with its own challenges and limitations. For example, the authors of [9] showed how the position of texts in pairwise comparisons can influence the outcomes of evaluation results when using GPT models. Other limitations are that LLMs can give higher scores to more verbose and long-winded sentences [10], and also prefer responses generated by themselves as opposed to other LLMs [11].

Another variation of Evaluation strategy still based on LLMs is represented by fine-tuning specialized, open-source models specifically for evaluation purposes. This pattern typically involves crafting high-quality Evaluation datasets (either synthetically with a powerful LLM, or through human curation), which are then used to fine-tune LLMs to try and distill the Human Evaluators' knowledge, as in [12].

In summary, for most use cases involving the generation of open-ended text where no ground truth is available the usual process involves combining some of the "automated" strategies mentioned above with Human-based evaluation, with varying weight given to the Automated vs Human evaluation based on the particular needs. When the tasks are broader, more nuanced, and "vague", or for tasks where a detailed explanation of the evaluation scores is needed, human evaluation is typically given more weight.

3. Methodology: Evaluation Framework for Survey Generation

Before diving deep into the architecture of our framework, we introduce and formalize a few basic concepts.

3.1. Background: Basic Concepts

A *survey* is a questionnaire used to collect data from a group of people to gather information, opinions, or feedback on a particular topic or subject. We formally introduce a *survey* as:

Definition 1 (Survey)**.** *A Survey typically consists of several questions designed to gather specific information from respondents. We define a survey as a tuple* $< h, l >$ *where* h *represents the survey title and* l *is the list of questions composing the survey.*

In turn, a single *survey question* can be defined as:

Definition 2 (Survey question)**.** *We define a Survey question as a tuple* $< t, k, o >$ *where* t *represents the survey question text,* k *is its type drawn from a predefined taxonomy* K*, and* o *represents the list of answer options. Examples of survey question types belonging to* K *include: open-ended questions, Net Promoter Score (NPS) questions, contact information questions, rating questions, and more. Except for the "Open-ended" questions, a survey question usually has a list of user-defined answer options amongst which the respondent may choose to respond to the question. For example, in the question "What's your work status?", possible answer options could be: "Employed", "Self-employed", "Interning", "Part-time", and "Unemployed".*

In our platform, users can leverage the *BWAI* feature to automatically generate surveys. This process involves users providing their survey intent through a written text (the prompt). Using LLMs, we can streamline the process and allow users to generate high-quality surveys with

minimal effort, increasing the level of our user experience. We formalize a user prompt as follows:

Definition 3 (User prompt)**.** *The User prompt embodies the user's intention when creating a survey. Through text, the user can articulate the desired structure and content of the survey.*

These generated surveys are designed to align with our established standards, which are the culmination of years of research on best practices and recommendations for creating surveys for large audiences. Our aim is to leverage our domain knowledge to help users to create high-quality surveys. To achieve this, we incorporate elements of our guidelines and best practices directly into the system prompt which defines the "behaviour" of the LLM.

Definition 4 (System prompt)**.** *The System prompt serves as the blueprint for instructing the LLM on generating surveys in accordance with elements of our established standards.*

Nevertheless, we acknowledge the challenge of ensuring that LLM models accurately follow our instructions, given their inherently unpredictable behavior. We formalize this problem as follows:

Definition 5 (Survey Generation Reliability Problem)**.** *Given a user prompt p_u, a system prompt p_s, and a generative model g, our objective is to automatically generate a survey s. The generated survey s should accurately reflect the user's intent as specified in p_u, while also adhering to the survey standards and guidelines detailed in p_s.*

3.2. Framework architecture

In order to continuously improve the quality of the surveys generated with *BWAI* we typically work in iterative cycles, which may introduce new issues while addressing existing ones, potentially impacting model quality. These issues can arise mainly due to changes in the prompts to accommodate new functionalities or due to switches and upgrades in the generative models at the core of the feature.

To mitigate this risk, we propose a testing framework with automatic tests to ensure expected model behaviors, aiding in risk assessment regarding survey standards and increasing our confidence when evaluating model updates. Unlike traditional machine learning problems such as classification or regression tasks which have well-defined test sets (ground truth), generative features lack this, necessitating such a framework. Its scope is not to monitor data, but to validate model functionality due to changes in the *BWAI* components. The ultimate goal is to maintain a reliable user experience for our customers, safeguarding against the deployment of new model versions that could introduce unforeseen behavior.

Figure 1 illustrates the comprehensive workflow implemented in our Survey Generation Testing Framework . The User prompts, as described in Definition 3, represent authentic prompts logged in our platform, conveying users' intentions for survey creation. Highlighted in blue are the pivotal components utilized by the *BWAI* tool for survey generation: the system prompt (as defined in 4) and the generative model (e.g., GPT models or open-source LLMs like llama, mistral, etc.). These components constitute the fundamental dimensions of our framework. Generated surveys are leveraged for metadata feature extraction and distributional drift tests. With varied settings of system prompts and generative models, the Survey Generation Testing Framework conducts pairwise analyses to discern drifts between these configurations. These steps are better detailed in the next two subsections.

3.3. Survey metadata features

To measure the impact of our developments on the outputs of the system, we define several metadata features that are computed on sets of surveys generated with different configurations of the *BWAI* feature.

All the metadata features used are based on some attributes of the surveys. In Table 1, we outline the complete set of metadata features used in our framework, along with some relevant information. The column first column indicates the name of the feature, while the second one specifies the aggregation function applied to the data.

For example, given a list of questions Q for a given survey, the feature *n_open_ended_questions* is defined as a simple *Count* which counts the number of "Open Ended" questions in a survey:

$$n_open_ended_questions = \sum_{q_i \in Q} \mathbf{1}\{\text{type}(q_i) = \text{"open_ended"}\}$$

Most of the features are based on numerical attributes of the survey, with the only exceptions being represented by the feature *any_special_character*, which is a Boolean attribute, and the features *dist_unigrams* and *dist_bigrams* which are both categorical attributes. One noteworthy feature is the *score_flesch_kincaid*, representing the Flesch-Kincaid Grade Level metric as defined in [13].

3.4. Distributional drift tests

We calculate the distribution for each metadata feature. To detect distributional shifts between two different configurations of system prompt and generative model for

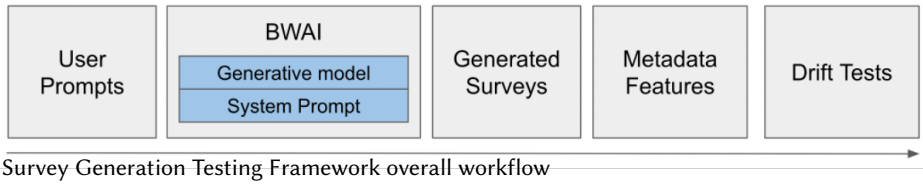

Figure 1: Survey Generation Testing Framework overall workflow

Feature Name	Aggregation
n_contact_info_questions	Count
n_open_ended_questions	Count
n_nps_questions	Count
n_multiple_selection_questions	Count
n_closed_ended_questions	Count
n_generated_questions	Count
n_unsupported_questions	Count
n_single_choice_questions	Count
n_characters_in_survey	Count
n_words_in_survey	Count
n_unsupported_questions	Count
std_n_words_per_question	Std
avg_word_length	Mean
avg_n_answer_options	Mean
avg_n_words_per_question	Mean
avg_n_words_per_answer_option	Mean
max_word_length	Max
any_special_character	Any
score_flesch_kincaid	Count
dist_unigrams	Count
dist_bigrams	Count

Table 1
List of survey metadata features supported by our framework

a specific metadata feature, we compute the Population Stability Index (PSI).

The PSI is a synthetic measure of how much a population has shifted over time or between two different samples of a population. It achieves this by categorizing the two distributions into buckets and assessing the percentage of items in each bucket, culminating in a single scalar value that indicates the disparity between the populations [14]. We use the popular PSI formula:

$$PSI = \sum_{i=1}^{n} (P_t^i - P_b^i) \cdot \ln\left(\frac{P_t^i}{P_b^i}\right)$$

Where:

- P_i^t is the proportion of the population in the i-th bin (or segment) at time t (typically the test or current time period).
- P_i^b is the proportion of the population in the i-th bin (or segment) at the baseline time period (typically the training or historical time period).
- n is the total number of bins (or segments) in the distribution.

The typical interpretations of PSI outcomes are as follows:

- PSI < 0.1: Indicates no significant population change.
- PSI < 0.2: Reflects a moderate population change.
- PSI $\geq$ 0.2: Signifies a significant population change.

For our framework, we use 0.2 as the threshold (λ) for the PSI score. Therefore, any value above the score is called a FAILED test, indicating significant changes in the distributions. For better clarity, Algorithm 1 presents the drift test function algorithm utilized in our framework.

Algorithm 1 Metadata drift test algorithm

```
procedure DRIFT_TEST(m, λ)
▷ Inputs:
m is a given survey metadata feature distribution.
λ is the PSI threshold.
    if PSI(m) > λ then
        return FAIL
    else
        return PASS
    end if
end procedure
```

4. Experimental results

4.1. Experiment setup:

The *BWAI* system is made up of two primary elements: (a) the system prompt and (b) the generative model. When the feature was released in late 2023 the first version was relying on GPT3.5-Turbo as the core LLM and used a version of the system (including prompts and logic) we will refer to as *v1*. Later we updated several components of the system, and we will refer to this updated version as *v2*. Also, we have experimented with GPT4-Turbo as a base LLM.

Given this context, in this paper we present real-case tests conducted using two system versions (*v1* and *v2*)

and two models for analysis: GTP-3.5 Turbo (GPT3.5) and GPT-4 Turbo (GPT4), both under the "0125" release from the OpenAI API.

BWAI configuration ($\mathscr{B}^{GenModel}_{SysPrompt}$). Our objective is to evaluate the differences in survey generation across various combinations of generative models (i.e. GPT3.5 and GPT4) and system prompt versions (i.e. *v1* and *v2*). We list all the pairs of evaluations that we focused on in this analysis. The idea is to have at least one common element in the tuple (i.e. either the prompt or the generative model) to assess the impact when transitioning between versions:

1. $<\mathscr{B}^{GPT3.5}_{v1}, \mathscr{B}^{GPT3.5}_{v2}>$
2. $<\mathscr{B}^{GPT3.5}_{v1}, \mathscr{B}^{GPT4}_{v1}>$
3. $<\mathscr{B}^{GPT3.5}_{v2}, \mathscr{B}^{GPT3.5}_{v2}>$
4. $<\mathscr{B}^{GPT4}_{v1}, \mathscr{B}^{GPT4}_{v2}>$

System prompts differences. Regarding the differences between the *v1* and *v2* system prompts, we summarize some of the key improvements that the v2 prompt introduces over the previous version:

- Addition of multilingual support
- Improved output formatting instructions
- Improved instructions to encourage the system to comply with survey research best practices (i.e. avoid open-ended questions where not necessary, order questions from general to specific, etc.)
- Addition of specific instructions to improve creativity
- Longer prompt with much more structure in the system prompt (416 to 690 tokens)
- Support for additional use cases such as survey forms

User prompts collection. In order to measure the differences across the system configurations outlined above we selected a subset of 3185 input prompts which have been collected from real customers who have interacted with the *BWAI* system and consented to let us use their prompts to improve our system. The selection has been done starting from the full set of user prompts collected between October 2023 and January 2024 and applying the following filters in sequence (with filtering boundaries and parameters determined through ad-hoc analyses to exclude poor quality samples):

1. Drop duplicates;
2. Drop input prompts which contain PII or sensitive information as flagged by our internal privacy-preservation pipelines;
3. Select only inputs written in English;
4. Drop inputs shorter than 200 characters and longer than 500 characters;
5. Drop inputs which led to generated surveys with an outlier number of questions (i.e. <5 or >12).

4.2. Drift tests: Overall results

Experiment	FAIL	PASS
$<\mathscr{B}^{GPT3.5}_{v1}, \mathscr{B}^{GPT3.5}_{v2}>$	5	16
$<\mathscr{B}^{GPT3.5}_{v1}, \mathscr{B}^{GPT4}_{v1}>$	13	8
$<\mathscr{B}^{GPT3.5}_{v2}, \mathscr{B}^{GPT4}_{v2}>$	16	5
$<\mathscr{B}^{GPT4}_{v1}, \mathscr{B}^{GPT4}_{v2}>$	9	12

Table 2
Overall results of drift tests

For each of the metadata features introduced in 1, we perform the distribution drift tests.

The number of passed and failed drift tests is shown in table 2. A failed test means that there is a drift in the output. As introduced in the section 3.4, it measures drift by using the Population Stability Index of the two distributions.

We observe a significant increase in the number of failed tests when transitioning from the GPT3.5 to the GPT4 model. Specifically, there were 16 failed tests when comparing these two models for the system prompt version *v2*, and 13 failures for version *v1*.

4.3. Drift tests: per feature

In this section, we conduct a detailed examination of the experiment results presented in the previous section, focusing on the analysis of actual drift scores of metadata features across the different *BWAI* configurations. Table 3 provides a comprehensive overview of the PSI drift scores for the metadata features. We focus only on the ones having at least a failed case.

One noteworthy case involves the metadata feature *n_contact_ info_questions*, which exhibits a PSI score of 3.778. The histograms for this metadata feature is shown in Figure 2. This indicates a significant drift primarily due to the transition of models (i.e., from GPT3.5 to GPT4), without any modifications in prompts where in both cases the *v1* system prompt was used. In turn, for *v2*, the highest drift was observed for the metadata feature *n_generated_questions* with PSI equals to 2.950. This happens when upgrading the generative model from GPT3.5 to GPT4.

When assessing changes induced solely by changes in the system prompts (i.e., transitioning from v1 to v2) while maintaining the same generative model, overall lower metadata drift scores are observed. Specifically, when utilizing the GPT3.5 model, the highest score among these cases was reported for the metadata feature *n_multiple_selection_questions* (0.642). Conversely, with the GPT4 model as the generative model, the highest score resurfaced for the metadata feature *n_contact_info_questions* (1.540). The histograms for this case are shown on Figure 3.

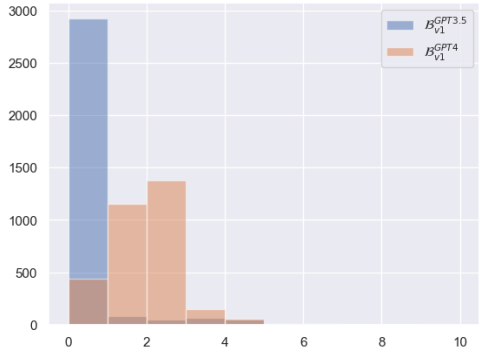

Figure 2: Histograms for the metadata feature ***n_contact_info_questions*** extracted from surveys generated using GPT4 and GPT3.5 with *v1* only

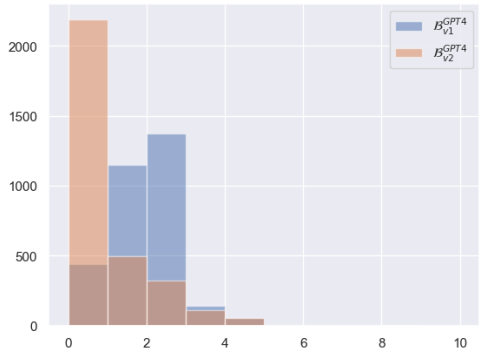

Figure 3: Histograms for the metadata feature ***n_contact_info_questions extracted*** from surveys generated using GPT4 with *v1* and *v2* prompts

In practice, this framework serves as a valuable tool for assessing whether intended modifications to system prompts translate effectively into the survey generation process. For instance, the updated system prompt includes specific instructions to nudge the LLM to generate questions which include answer options (as opposed to open-ended questions which do not). One of these question types is represented by the Multiple Selection question type, and the impact of these instructions between V1 and V2 can be seen on the scores in Table 3 on the row corresponding to the feature *n_multiple_selection_questions*, as well as in Figure 4 where it is clearly shown that the V2 prompt tends to generate more *multiple_selection* questions than the V1 prompt. Also, through the detection of distributional drift of the survey metadata features, we can identify and mitigate potential issues, thereby avoiding unexpected behaviors of the feature.

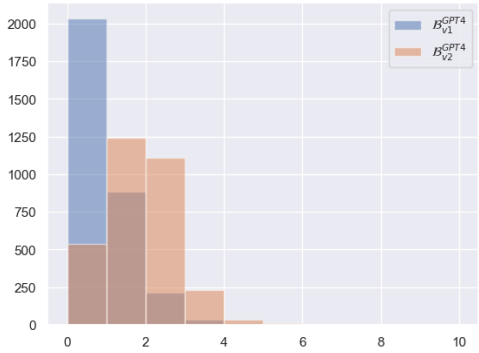

Figure 4: Histograms for the metadata feature ***n_multiple_selection_questions*** extracted from surveys generated using GPT4 with *v1* and *v2* prompts

5. Conclusion

In this study, we proposed a comprehensive evaluation framework to enhance the reliability of Large Language Model (LLM)-based systems for survey generation tasks. By addressing the challenges associated with accurately following user prompts and maintaining consistency with established standards, the framework functions as a protective barrier, effectively setting guardrails to preempt unforeseen behaviors of our BWAI tool. Through the detection of distributional drift of the survey metadata features, the framework acts as a guiding compass for data scientists to investigate and address any unintended deviations in the application's behavior, thereby ensuring its stability and reliability.

Our experimental results demonstrate the effectiveness of the proposed framework in evaluating survey generation metadata features across different configurations of system prompts and generative models. We observed significant differences in survey outputs when transitioning between different versions of LLM models, highlighting the importance of comprehensive evaluation in adapting to model updates. Furthermore, our analysis revealed nuanced insights into the impact of system prompt versions on survey generation quality, underscoring the need for careful consideration of both prompt design and model selection in ensuring reliable survey generation.

As future work, we aim to integrate automated evaluation strategies to assess the "quality" of the generated surveys. In this scenario, the emphasis shifts from leveraging metadata features to compare differences across different system versions to analyzing the survey content itself. One promising direction is to use LLMs to act as preliminary inspectors of survey quality. This could significantly accelerate our quality assessment process, which currently relies heavily on human evaluation.

Features	$<\mathscr{B}_{v1}^{GPT3.5}, \mathscr{B}_{v2}^{GPT3.5}>$	$<\mathscr{B}_{v1}^{GPT3.5}, \mathscr{B}_{v1}^{GPT4}>$	$<\mathscr{B}_{v2}^{GPT3.5}, \mathscr{B}_{v2}^{GPT4}>$	$<\mathscr{B}_{v1}^{GPT4}, \mathscr{B}_{v2}^{GPT4}>$
avg_n_answer_options	0.405	0.478	0.424	0.351
avg_n_words_per_answer_option	0.642	0.311	0.339	0.622
avg_n_words_per_question	0.000	0.526	0.392	0.367
drift:bigrams_distribution	0.565	0.860	0.795	0.673
drift:unigrams_distribution	0.205	0.247	0.222	0.217
max word length	0.000	0.000	0.219	0.000
n_closed_ended_questions	0.000	0.318	1.203	0.447
n_contact_info_questions	0.000	**3.778**	0.529	**1.540**
n_generated_questions	0.000	2.624	**2.950**	0.000
n_multiple_selection_questions	**0.991**	0.000	0.395	1.271
n_nps_questions	0.000	1.899	2.300	0.000
n_open_ended_questions	0.000	0.352	0.846	0.000
n_single_choice_questions	0.000	0.000	0.504	0.000
n_words_in_survey	0.000	1.535	2.068	0.000
std_n_words_per_question	0.000	1.173	0.419	0.599
n_characters_in_survey	0.000	1.446	1.871	0.000

Table 3
PSI scores per feature. We show only the features that had failed in at least one of the BWAI configurations ($\mathscr{B}$).

References

[1] C. Gómez-Rodríguez, P. Williams, A confederacy of models: a comprehensive evaluation of llms on creative writing, 2023. arXiv:2310.08433.

[2] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, B. Newman, B. Yuan, B. Yan, C. Zhang, C. Cosgrove, C. D. Manning, C. Ré, D. Acosta-Navas, D. A. Hudson, E. Zelikman, E. Durmus, F. Ladhak, F. Rong, H. Ren, H. Yao, J. Wang, K. Santhanam, L. Orr, L. Zheng, M. Yuksekgonul, M. Suzgun, N. Kim, N. Guha, N. Chatterji, O. Khattab, P. Henderson, Q. Huang, R. Chi, S. M. Xie, S. Santurkar, S. Ganguli, T. Hashimoto, T. Icard, T. Zhang, V. Chaudhary, W. Wang, X. Li, Y. Mai, Y. Zhang, Y. Koreeda, Holistic evaluation of language models, 2023. arXiv:2211.09110.

[3] Z. Xie, T. Cohn, J. H. Lau, The next chapter: A study of large language models in storytelling, 2023. arXiv:2301.09790.

[4] Z. Shao, M. Huang, J. Wen, W. Xu, X. Zhu, Long and diverse text generation with planning-based hierarchical variational model, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3257–3268. URL: https://aclanthology.org/D19-1321. doi:10.18653/v1/D19-1321.

[5] J. Li, M. Galley, C. Brockett, J. Gao, B. Dolan, A diversity-promoting objective function for neural conversation models, in: K. Knight, A. Nenkova, O. Rambow (Eds.), Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 110–119. URL: https://aclanthology.org/N16-1014. doi:10.18653/v1/N16-1014.

[6] Y. Zhu, S. Lu, L. Zheng, J. Guo, W. Zhang, J. Wang, Y. Yu, Texygen: A benchmarking platform for text generation models, 2018. arXiv:1802.01886.

[7] W. Yuan, G. Neubig, P. Liu, Bartscore: Evaluating generated text as text generation, 2021. arXiv:2106.11520.

[8] M. Gao, X. Hu, J. Ruan, X. Pu, X. Wan, Llm-based nlg evaluation: Current status and challenges, 2024. arXiv:2402.01383.

[9] J. Wang, Y. Liang, F. Meng, B. Zou, Z. Li, J. Qu, J. Zhou, Zero-shot cross-lingual summarization via large language models, 2023. arXiv:2302.14229.

[10] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, I. Stoica, Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. arXiv:2306.05685.

[11] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, C. Zhu, G-eval: Nlg evaluation using gpt-4 with better human alignment, 2023. arXiv:2303.16634.

[12] W. Xu, D. Wang, L. Pan, Z. Song, M. Freitag, W. Y. Wang, L. Li, Instructscore: Explainable text generation evaluation with finegrained feedback, 2023. arXiv:2305.14282.

[13] J. Kincaid, Derivation of New Readability Formulas: (automated Readability Index, Fog Count and Flesch

Reading Ease Formula) for Navy Enlisted Personnel, Research Branch report, Chief of Naval Technical Training, Naval Air Station Memphis, 1975. URL: https://books.google.it/books?id=4tjroQEACAAJ.

[14] R. Taplin, C. Hunt, The population accuracy index: A new measure of population stability for model monitoring, Risks 7 (2019) 53.