In our work, we randomly sampled 12K questions from the train set of each of these datasets paired with their provided answers. After removing questions that were less than 4 words or more than 40 words, we were able to obtain 11,469 unique questions and their respective answers from these datasets. We split these questions using a 70-30 ratio for training and testing, i.e., 8,028 and 3,441 questions, respectively. We consider each of these questions to be prompts that would be submitted to an LLM for a response.

Generating Prompt Variations. For each of the 11,469 unique prompts in our dataset, we generate several prompt variations. The objective of this step to generate slight variations of the original prompt, each of which may or may not be satisfiable by the LLM. To generate prompt variations, we utilize two widely-used LLMs that have demonstrated strong performance across many downstream tasks, namely the pre-trained LLaMA 3.1 with 8B parameters [28] and Mistral-nemo [20]. We chose open-source models to ensure reproducibility and facilitate further research in *prompt sensitivity*. We designed the instructions for the LLMs to be as clear and straightforward as possible, similar to those intended for human use. The primary goal was to generate variations that retain the same semantics of the original prompt, instructing the model to produce a rephrased prompt that does not answer the question directly but maintains the same information need and semantic content [*]. However, while instructed explicitly, LLMs occasionally deviate from the instructions due to hallucination. To address this, we filter out prompts that did not have at least nine valid variations generated and excluded prompts with fewer than four terms to maintain quality and consistency across the dataset. At the end, our dataset consists of 11,469 prompts, each with 9 different variations, resulting in 114,690K variations. We ran each of these prompts (the original prompt and its nine variations) against the LLMs and generated an answer for each. We then compared the produced answer against the expected answer in the TriviaQA and HotpotQA datasets. Each prompt or prompt variations were labeled as being answerable by the LLMs depending on the answer they generated and whether it aligned with the expected answer. On this basis, the objective of the *Prompt Sensitivity Prediction* task is to predict whether an LLM can correctly answer an input prompt.

Establishing Baselines. To benchmark this task, we identify three types of tasks from the literature that may be applicable to prompt sensitivity prediction:

(1) We employ LLMs directly by asking them to self-assess their ability to predict whether they can accurately answer a given prompt or not [38]. This approach, which we refer to as the LLM self-evaluation baseline, relies on the model's internal confidence in its responses. The prompts and instructions used for this baseline can be found in our GitHub repository for reproducibility.

(2) We further treat prompt sensitivity prediction as a text classification task. We train a text classifier on our dataset's training set and evaluate it on the test set to predict whether the LLM's response to a prompt will meet users' information need. We used the text classifiers implemented by Rajapakse et al. [33].

(3) Prompt sensitivity prediction is also conceptually related to the task of query performance prediction [31,34,2] whose goal is to estimate the quality of retrieved documents in response to a user query. For our task, we adapt QPP methods to predict whether a prompt will yield a correct response from an LLM or not. QPP methods

[*] Due to space constraints, we have provided the full set of instructions in our GitHub repository.

Table 2: Results of baselines on PromptSET.

	Category	Method	PromptSET-TriviaQA				PromptSET- HotPotQA			
			Accuracy	F1	Recall	Precision	Accuracy	F1	Recall	Precision
Mistral Answers	LLM-Based	Mistral	0.5045	0.5858	0.7743	0.4711	0.3735	0.2005	0.6912	0.1173
		LLaMA	0.4656	0.6239	0.9798	0.4577	0.1696	0.2050	0.9419	0.1150
	Text Classification	BERT	0.660	0.659	0.620	0.654	0.526	0.360	0.017	0.813
	Specificity-based QPP	CC	0.506	0.453	0.452	0.454	0.549	0.209	0.524	0.130
		DC	0.484	0.448	0.463	0.434	0.565	0.199	0.475	0.126
		IEF	0.505	0.462	0.469	0.455	0.535	0.204	0.526	0.127
		PageRank	0.481	0.444	0.458	0.431	0.533	0.153	0.370	0.096
	Supervised QPP	BERTPE	0.648	0.627	0.644	0.611	0.710	0.318	0.594	0.217
LLaMA Answers	LLM-Based	Mistral	0.5160	0.6045	0.7704	0.4974	0.3731	0.1978	0.6930	0.1153
		LLaMA	0.4940	0.6507	0.9818	0.4866	0.1674	0.2013	0.9408	0.1127
	Text Classification	BERT	0.664	0.664	0.651	0.650	0.532	0.377	0.034	0.808
	Specificity-based QPP	CC	0.500	0.463	0.449	0.478	0.545	0.199	0.507	0.123
		DC	0.484	0.464	0.465	0.463	0.562	0.190	0.462	0.120
		IEF	0.510	0.482	0.475	0.489	0.535	0.202	0.529	0.125
		PageRank	0.482	0.461	0.461	0.461	0.534	0.151	0.371	0.094
	Supervised QPP	BERTPE	0.659	0.651	0.646	0.656	0.710	0.314	0.596	0.213

typically fall into pre-retrieval and post-retrieval categories. However, since generative settings do not produce traditional “retrieved lists”, only pre-retrieval QPP methods are applicable in our context. Furthermore, collection-dependent QPP methods [18] are also not applicable due to their dependence on a document corpus which is not available when using an LLM for generating a response. Thus, we adopted BERT-PE [23], a SOTA pre-retrieval QPP model that similar to [35,13], uses contextualized embeddings to learn query performance. Additionally, we considered the neural embedding specificity-based QPP metrics to assess prompt sensitivity [9,8,7,1,6], namely Closeness Centrality (CC), Degree Centrality (DC), PageRank , and Inverse Edge Frequency (IEF), all measured on the ego network of the query terms in the embedding space. We note that since the output of QPP methods is a scalar value, inspired by previous studies [14,12,26], we convert them to binary by classifying values above and below the mean of the data.

3 Experiments and Findings

Baseline Performance. We analyze the performance of the of baselines on the PromptSET test set. Table 2 presents the results for predicting whether each prompt variation could be correctly answered using the LLaMA and Mistral. We report results in terms of accuracy, F1, recall, and precision. As shown in the table, the specificity-based QPP methods (i.e., CC, DC, IEF, and PageRank) perform the lowest among the baselines. Since QPP methods were not specifically designed for prompt sensitivity prediction, their performance is relatively weak on both datasets. We hypothesize that the specificity levels across the original prompts and their variations are too similar, making it challenging for specificity metrics to effectively distinguish between different levels of prompt specificity. On the other hand, BERT-PE demonstrates higher effectiveness in determining whether a prompt can be answered correctly. BERT-PE, which is supervised QPP method, shows

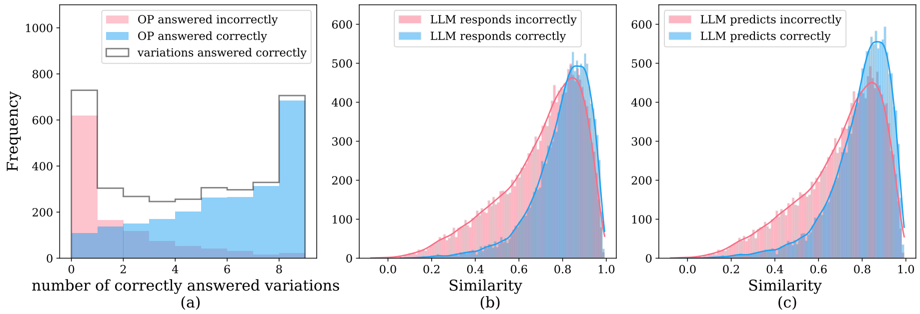

Fig. 1: (a) A histogram of correctly answered variations when the original prompt yields a correct response (in blue) or an incorrect response (in red), along with the total count of correctly answered variations (black step). Figures (b) and (c) display histograms for correct (blue) and incorrect (red) responses and predictions, respectively.

competitive performance to text classification-based methods on PromptSET-TriviaQA and also outperforms other baselines significantly on PromptSET-HotpotQA. This suggests that *supervised* QPP methods might be well-suited for the prompt sensitivity prediction task. We finally note the performance of LLM self-evaluation baseline. This baseline shows reasonable performance on TriviaQA but lacks consistency on HotpotQA. This is inverse to the performance of BERT-PE, indicating that these methods do not show stable performance across different prompt subsets.

Impact of original prompt correctness on variation answerability[*]**.** Here, we aim to investigate whether an LLM's ability to answer the original prompt influences its ability to answer other variations. Specifically, if the LLM can answer the original prompt, does this increase the likelihood of correctly answering the variations as well? To this end, Figure 1(a), marked with a black line, presents histograms of prompts showing the frequency of correctly answered variations out of a total of 10. In this figure, the x-axis represents the count of correctly answered variations for each prompt, grouping prompts by the number of successful variations. The distribution appears roughly balanced rather than long-tailed, indicating that PromptSET includes prompts with diverse answerability across reformulated variations, from those with only one answerable variation to those where all variations are answerable. In addition, in Figure 1(a), we further break down the results based on whether the original prompt was answered correctly or not. We observe that when the original prompt is answered correctly (shown in the blue histogram), there is a higher number of variations that also yield correct answers, reflected by an ascending pattern in the histogram. Conversely, when the original prompt is answered incorrectly, most of the prompts have only one out of the 10 variations answered correctly, displaying a descending pattern in the red histogram.

Impact of similarity to the original prompt. Here, we aim to investigate the impact of variation similarity to the original prompt on the predictability of LLM responses. Specifically, we ask whether a variation that is more similar to the original prompt has a higher likelihood of generating a correct answer, while less similar variations may lead to lower predictability. To test this hypothesis, we conducted experiments, as

[*] Due to space constraints, we report the full findings in our GitHub repo.