scenario	train	test	scenario	train	test	scenario	train	test
others	317	79	writing_cooking_recipe	40	11	classification_identification	24	6
functional_writing	128	32	explaining_code	40	10	language_polishing	22	4
brainstorming	90	24	writing_legal_document	40	10	chitchat	22	7
seeking_advice	88	25	asking_how_to_question	40	10	writing_product_description	20	5
open_question	77	20	writing_presentation_script	38	10	data_analysis	18	5
explaining_general	66	17	writing_social_media_post	38	10	writing_marketing_materials	17	5
instructional_rewriting	58	15	question_generation	38	10	note_summarization	17	4
verifying_fact	49	13	planning	38	10	paraphrasing	17	5
analyzing_general	49	13	writing_blog_post	36	9	writing_technical_document	17	5
title_generation	48	12	writing_job_application	36	10	text_simplification	16	5
code_generation	48	12	writing_personal_essay	36	10	information_extraction	16	2
roleplay	47	12	value_judgement	35	9	writing_biography	16	4
rejecting	45	12	code_to_code_translation	32	9	text_correction	12	6
creative_writing	45	12	writing_advertisement	31	8	reading_comprehension	12	3
exam_question_without_math	44	12	writing_email	30	8	keywords_extraction	12	3
writing_song_lyrics	44	11	recommendation	29	8	topic_modeling	10	3
text_to_text_translation	43	11	ranking	28	8	writing_scientific_paper	10	3
text_summarization	43	12	counterfactual	26	7	peer_review	7	2
code_correction_rewriting	43	11	exam_question_with_math	24	4	code_simplification	6	2
math_reasoning	41	12	writing_news_article	24	6	overll	2383	623

Table 7: The scenario distribution in the training and test set for scenario classifier, note that “rejecting” and “peer_review” are two early-defined scenarios that have been removed by us.

B Training Details of Scenario Classifier

In this section we describe in detail the training process of the scenario classifier mentioned in §3.2.

We model the scenario classification task as a generation task. The classifier are required to generate only the scenario name when given the query, with the prompt as `"Identify the scenario for the user's query, output 'default' if you are uncertain.\n\nQuery:\n\n{input}\n\nScenario:"` (the "default" scenario in the prompt is the early naming for "others" scenario).

In general, the training involves three steps:

1. We first brainstorm about 10 seed queries for each scenario with the help of ChatGPT, and train a model that can directly output the scenario name when given a query as a conditional generation task on this small synthetic dataset.
2. Using the trained model, we conducted an initial classification for queries in Chatbot Arena Conversations and ShareGPT[4] as they cover much more scenarios than other datasets. Based on this preliminary classification, we randomly select up to 50 queries from each scenario for a secondary manual validation, involving data cleaning and correcting misclassified labels.
3. We combine the newly-collected dataset and the small synthetic dataset in step 1, and retrain our final classifier. We divide queries in each scenario in an 8:2 train/test split (Tab. 7). The accuracy and F1 of the final classifier on test set are 72.55 and 74.12, respectively.

Our scenario classifier is trained from LLaMA-2-13B (Touvron et al., 2023b), and we set the max sequence length as 2,048, and the max length for query as 2,048-50=1,998 both in training and inference. If a query Q with length L exceeds that limit, we truncate it from the middle and replace the dropped part with a "..." since the front and end of the sequence usually contain more important information for identifying scenario of the (such as the user's instruction): $Q_{1:L} \rightarrow [Q_{1:999}; ...; Q_{L-1000:L}]$.

We train the scenario classifier for 3 epochs on the training set, and set the batch size as 64. Without warmup steps, we set the initial learning rate to 1e-5 and cosine decaying to 0 by the end of training. The optimizer is AdamW with $\beta_1 = 0.9, \beta_2 = 0.95$ as in training AUTO-J, and we also use the speedup and GPU memory saving techniques like DeepSpeed Zero 3, BF16, TF32, and gradient-checkpointing. The loss is only calculated on the output end as well.

[4]This dataset is collected from `https://sharegpt.com/` containing shared conversations with ChatGPT or GPT-4. We use a public available subset of it.