Automatic Generation of Model and Data Cards: A Step Towards Responsible AI

Anonymous ACL submission

Abstract

In an era of model and data proliferation in machine learning/AI especially marked by the rapid advancement of open-sourced technologies, there arises a critical need for standardized consistent documentation. Our work addresses the information incompleteness in current human-generated model and data cards. We propose an automated generation approach using Large Language Models (LLMs). Our key contributions include the establishment of CARDBENCH, a comprehensive dataset aggregated from over 4.8k model cards and 1.4k data cards, coupled with the development of the CARDGEN pipeline comprising a two-step retrieval process. Our approach exhibits enhanced completeness, objectivity, and faithfulness in generated model and data cards, a significant step in responsible AI documentation practices ensuring better accountability and traceability.[1]

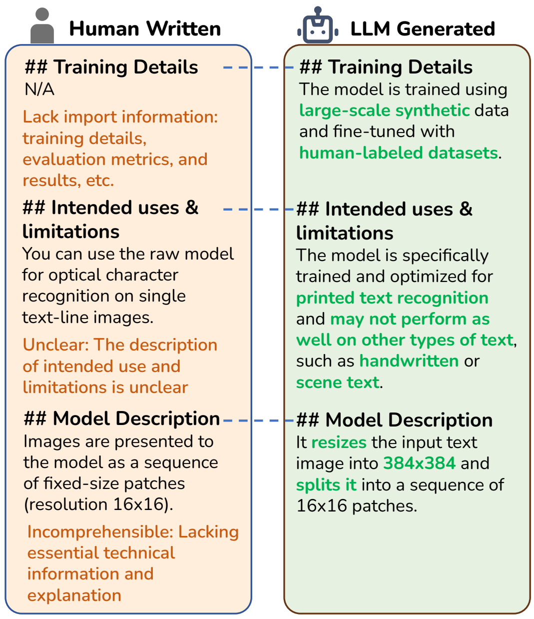

Figure 1: Common problems with manually generated model cards and data cards.

1 Introduction

The landscape of artificial intelligence (AI) has undergone a profound transformation with the recent surge in open-sourced models (Villalobos et al., 2022; Sevilla et al., 2022) and datasets (Northcutt et al., 2021; Sevilla et al., 2022). The trend has been significantly accelerated by the advent of disruptive technologies such as transformers (Gruetzemacher and Whittlestone, 2022; Vaswani et al., 2017). Since this proliferation of accessible models and datasets can have their applications significantly influence various aspects of society, it becomes increasingly important to underscore the necessity for standardized consistent documentation to communicate their performance characteristics accurately (Liang et al., 2022).

In this context, model cards proposed by Mitchell et al. (2019) and data cards proposed by Pushkarna et al. (2022), emerge as necessary documentation tools. These cards bridge the communication gap between model/data creators and product developers, thereby ensuring a comprehensive understanding of the model's/data's capabilities and limitations for both in academia as well as industrial applications (Pushkarna et al., 2022; Sevilla et al., 2022; Vaswani et al., 2017; Sevilla et al., 2022). Model/data cards are instrumental in research, offering detailed insights such as data characteristics, sources, etc, as well as model architecture, training procedures, and potential biases and limitations, which accelerates development and reduces error propagation in subsequent models (Swayamdipta et al., 2020).

Inspired by these concepts, HuggingFace (HF) developed card specifications for models and datasets hosted on its website. Despite the release of some

[1] Our code and data have been uploaded to the submission system, and will be open-sourced upon paper acceptance.

available tools to assist model card writing[2], HF leaves the decision of what to report up to developers. This raises several problems: First, this approach relies heavily on the developers' understanding and interpretation of what should be reported, leading to inconsistencies and potential omissions of critical information (Shukla et al., 2021). Second, there is a tendency among card creators to use completed cards as templates rather than starting from the standardized template provided (Pushkarna et al., 2022). Such variability compromises the comprehensiveness and reliability of the cards.

With the power of state-of-the-art LLMs (Touvron et al., 2023; Brown et al., 2020; Ouyang et al., 2022; Jiang et al., 2023; Touvron et al., 2023), automatic generation of model and data cards can be served as an approach to ensure uniformity, consistency, and thoroughness across different model/data cards. To that end, we contribute the following: (1) A novel pioneering initiative to systematically utilize LLMs for automatically generating model/data cards; (2) CardBench, a curated dataset that encompasses all the associated papers and GitHub READMEs referenced in 4.8k model cards and 1.4k data cards; (3) A novel approach that decomposes the card generation task into multiple sub-tasks, proposing a CardGen pipeline including a two-step retrieval process; (4) A novel set of quantitative and qualitative evaluation metrics. We demonstrate that using our pipeline with GPT3.5, we achieve higher scores than human generated cards on completeness, objectivity, and understandability, demonstrating the effectiveness of CardGen pipeline.

2 Related Work

2.1 Accountability and Traceability for AI Systems Through Documentation

The increasing complexity of AI systems have raised significant concerns about their potential biases and non-transparency, thereby the negative implications for users and society (Jacovi et al., 2021; Barocas and Selbst, 2016; Panch et al., 2019; Daneshjou et al., 2021; Huang et al., 2023). This motivated the emergence of various documentation frameworks for ML models and datasets:

[2] https://huggingface.co/spaces/huggingface/Model_Cards_Writing_Tool

Model Cards Mitchell et al. (2019) proposed the concept of model cards as a framework for transparent documentation of machine learning models (ML) and provided detailed evaluations across diverse demographic groups and conditions. Advancements in model card design including the advocate of consumer labels' generation for ML models (Seifert et al., 2019), the principle introduction for explainable models (Phillips et al., 2020), other cards as complimentary to model cards (Adkins et al., 2022; Shen et al., 2021), environmental and financial impact considerations (Strubell et al., 2019), and some toolkits that help to track and report specific information in ML models (Arya et al., 2019; Shukla et al., 2021).

Data Cards In ML dataset documentation, Gebru et al. (2021) initiated datasheets for datasets, followed by the introduction of data statements for NLP data (Bender and Friedman, 2018; Bender et al., 2021), and data nutrition labels for better decision-making (Holland et al., 2020). McMillan-Major et al. (2021); Hutchinson et al. (2021) provided comprehensive data card templates. Pushkarna et al. (2022) proposed data cards for responsible AI development. Díaz et al. (2022) introduced CrowdWorkSheets for transparent documentation of crowdsourced data.

2.2 Knowledge-Enhanced Text Generation

LLMs can be augmented with external knowledge sources to improve their reasoning capabilities (Lewis et al., 2020; Li et al., 2022). Retriever, generator, and evaluator are the key components in a standard RAG system. With the advancement of powerful pretrained seq2seq models as generators, numerous studies have concentrated on retrieval and evaluation performance:

Dense Retrieval Dense retrievers match relevant contents with fully learned embeddings (Cao and Xiong, 2018; Lee et al., 2019), capturing more semantically similar texts than sparse retrievers using lexical overlaps (Robertson and Zaragoza, 2009). Pretrained retrieval representations have also been explored and used for zero-shot semantic matching (Reimers and Gurevych, 2019a; Gao and Callan, 2021; Günther et al., 2023; Lin et al., 2023). Researchers studied the transfer learning abilities (Thakur et al., 2021; Yu et al., 2022), using neural generative models as search indices (Metzler et al., 2021), and generating hypothetical documents before retrieval (Gao et al., 2023).

RAG Text Generation Evaluation Due to variations in retrieved content, customized generation pipelines, and user intentions, evaluating the effectiveness of LLM generated texts in a Retrieval-Augmented Generation (RAG) system becomes challenging (Huang et al., 2023; Mialon et al., 2023). Traditional n-gram based metrics like BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and PARENT-T (Wang et al., 2020b) are used for assessing the overlap between generated texts and references, but cannot fully grasp the quality nuances of human expectations (Honovich et al., 2021; Maynez et al., 2020). Some model-based metrics have later been invented to align better with human judgments without supervision, such as BERTScore (Zhang et al., 2019), MoverScore (Zhao et al., 2019), and BARTScore (Yuan et al., 2021). Research focused mainly on factuality (Gou et al., 2023; Chen et al., 2023; Galitsky, 2023; Min et al., 2023), and faithfulness (Barrantes et al., 2020; Fabbri et al., 2022; Santhanam et al., 2021; Laban et al., 2023; Durmus et al., 2020) of generation quality. Some frameworks have been designed to automate the assessment pipeline with the power of LLMs (Es et al., 2023; Pietsch et al., 2020; Liu et al., 2023; Fu et al., 2023; Manakul et al., 2023).

3 Defining the Model/Data Card Generation Task

3.1 Task Formulation

Denote our test set as $\boldsymbol{D} := \{(\boldsymbol{m}_i, \boldsymbol{p}_i, \boldsymbol{g}_i)\}_{i=1}^{N}$ consisting of N triples, each with a human-generated model card $\boldsymbol{m}_i$, a direct paper document $\boldsymbol{p}_i$, and a direct GitHub README document $\boldsymbol{g}_i$. For each question $\boldsymbol{q}_j$ from the question template set $\boldsymbol{Q} := \{\boldsymbol{q}_j\}_{j=1}^{M}$, we define a two-stage retrieve-and-generate task f_1 and f_2.

The retrieval task $f_1 : \mathcal{P} \times \mathcal{G} \times \mathcal{Q} \rightarrow \mathcal{R}$ maps source paper and GitHub documents according to the question to a set of retrieved chunks $\boldsymbol{R}$.

The generation task $f_2 : \mathcal{R} \times \mathcal{Q} \rightarrow \mathcal{A}$ maps the retrieved chunk set and questions to a space $\mathcal{A}$ that contains generated answers for all questions.

3.2 Structured Generation

Inspired by the model card design from Mitchell et al. (2019), HF provides its guidelines about how to fully fill out a model card.[3] It suggests a detailed disclosure of the model features and limitations in a published model card. Following the guidelines, we define seven sections including 31 individual questions for generating a complete model card. These sections are model summary, model details, uses, bias and risks, training details, evaluation, and additional information about the proposed model. We release our full question template for model cards and data cards in Appendix A. Table 1 shows the most important questions for each section of the full template.

4 CardBench Dataset

CardBench contains 4,829 human-generated model cards and 328 data cards with paper and GitHub references.

4.1 Dataset Collection

Data Source and Preprocessing We identify the model page[4] and the dataset page[5] of HF as data sources. We crawl the 10,000 most downloaded model cards (READMEs) and 10,000 most downloaded data cards from the HF page up to October 1, 2023. For each collected model card, we use regular expressions to find all valid paper URLs and GitHub repository URLs for both model cards and data cards. We leverage the SciPDF Parser[6] to parse downloaded paper PDFs into a JSON formatted data structure for the paper sections. We further use the GitHub REST API[7] to obtain README files of each repository. For each collected data card, we devise regular expressions to locate all data cards with the "Dataset Description" section, which should contain information such as the dataset homepage, paper link, and GitHub repository. Then, based on the information obtained from the data card, we retrieve and process paper documents and GitHub READMEs as done for model cards.

Evaluation Set Construction In the absence of standardized and strict content requirements by HF, collected model cards are mostly incomplete, and some examples are even minimally modified copies of existing ones. This variability undermines the

[3] https://huggingface.co/docs/hub/model-card-annotated
[4] https://huggingface.co/models
[5] https://huggingface.co/datasets
[6] https://github.com/titipata/scipdf_parser
[7] https://docs.github.com/en/rest?api

Question	Role	Prompt
Summary	Project organizer	Provide a 1-2 sentence summary of what the model is.
Description	Project organizer	Provide basic details about the model. This includes the model architecture, training procedures, parameters, and important disclaimers.
Direct use	Project organizer	Explain how the model can be used without fine-tuning, post-processing, or plugging into a pipeline. Provide a code snippet if necessary.
Bias, risks, limitations	Practical Ethicist	What are the known or foreseeable issues stemming from this model? These include foreseeable harms, misunderstandings, and technical and sociotechnical limitations.
Results summary	Developer	Summarize the model evaluation results.

Table 1: Template of the most important questions for each section.

reliability of our comparative evaluation against human-generated model cards as a reference metric. In an attempt to mitigate this shortcoming, we curate the highest quality human generated model cards to serve as our evaluation data set. This set comprised a select 350 examples that are rewritten by the HF team with their unique disclaimers. Also, for data cards, the majority of those collected are incomplete and lack content readability. In order to have a sufficient number of evaluation sets, we first selected all the data cards with a "Dataset Description" section. We then wrote markdown matching logic to obtain 300 examples as our evaluation set based on the word count and the number of sections in the data cards. See Appendix B for more details on data collection.

4.2 Data Annotation

In our methodology for generating model cards, emphasis is placed predominantly on the design details of the model itself, as opposed to referencing external methodologies being cited in human-generated model cards. It necessitates the identification of the primary paper proposing the model, along with the direct repository reflecting model implementation. The evaluation set is annotated by two ML Master's student researchers who know HF models well and are proficient in English. The process resulted in 294 evaluation examples having both direct paper and repository links. Additionally, to annotate the whole dataset, we prompt `GPT-3.5-Turbo` (Brown et al., 2020) to validate direct source document links, given the context wherein each URL is situated in the model card. We finally obtained 4,829 non-empty ones with either direct paper links or repository links. GPT's annotation reached 98.01% accuracy according to human validation results on the test set. For data cards, their primary paper link and direct repository responsible for the dataset is within the 'Dataset Description' section. We finally obtained 865 data cards with either direct paper links or repository links. This gain resulted in 99.7% accuracy according to human validation results on the 300 data cards test set. See Appendix C for human annotation guidelines and prompts for GPT validation.

	Split	Paper		GitHub	
		# Sections	# Words	# Sections	# Words
ModelCard	all	29	6810	22	2495
	test	30	6674	17	1855
DataCard	all	25	5741	9	975
	test	25	5784	8	816

Table 2: Statistics for direct paper documents and repository READMEs for crawled model cards and data cards, in terms of the average number of sections and the average number of words of documents.

4.3 Data Statistics

We show the overall statistics in Tables 2 and 11. We can observe that our test set, the set of model cards rewritten by the HF team, are more concise than other developer-written ones. Their corresponding source documents have similar sizes in terms of the number of sections and words.

To explore whether our test set represents the whole dataset well, we look into some model card features obtained with the HF API. Figure 8 shows that test set examples are nearly uniformly distributed compared to the overall dataset in terms of the number of downloads, and task distributions of models/datasets. A comparison of the test set to the whole set is shown in Figures 6 and 7. See Appendix D for additional dataset analyses.

5 Method: the CARDGEN Pipeline

5.1 Overview

Figure 2 shows our CARDGEN pipeline. For each $\boldsymbol{q}_j$ in $\boldsymbol{Q}$, we first prompt LLMs to split $\boldsymbol{q}_j$ into a sub-question set. Next, we use LLMs to infer relevant sections as potential knowledge sources, and generate pseudo answers for each sub-question leverag-

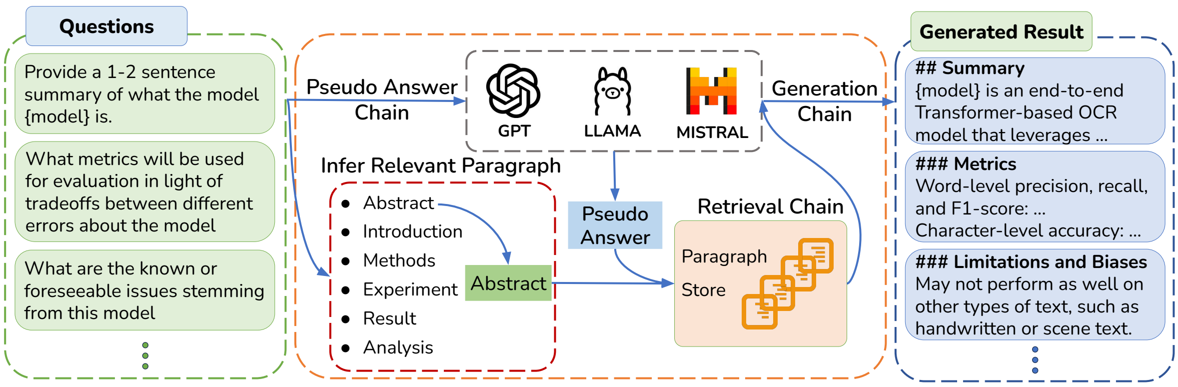

Figure 2: Overview of the CARDGEN pipeline to generate a full model card or a full data card.

ing LLM's own knowledge (Gao et al., 2023). The pseudo answer is used as a query to get the set $\boldsymbol{R}$ of relevant document chunks. We use an LLM to generate answers for the question prepended with highest-ranked document chunks.

5.2 Designing the Retriever

As the process of supervised retrieval necessitates the acquisition of additional crowd-sourced annotations for establishing ground truth sentences for each query, it constitutes a substantial amount of labor. Consequently, we choose to modify the standard RAG retrieval baselines (Lewis et al., 2020), where source documents are ranked based on the inner product similarity with a query question. We develop a two-step retrieval method to improve the retrieval precision: (1) Given all section names of a model's paper and README documents, we prompt the LLM to infer the top-k most plausibly relevant sections. (2) We query the pseudo answer from chunks in the inferred section contents after feeding it into an embedding model. We use the embedding model `jina-embeddings-v2-base-en` developed by Günther et al. (2023). This choice is further verified in Section 7.2.

5.3 Designing the Generator

For our CARDGEN pipeline, we test `GPT-4-Turbo` (OpenAI, 2023), `GPT-3.5-Turbo` (Brown et al., 2020), `Llama2 70B Chat` (Touvron et al., 2023), `Llama2 7B Chat` (Touvron et al., 2023), `Mistral 7B Instruct` (Jiang et al., 2023) as backbone LLMs. We generate the answer $\boldsymbol{t}_j$ to each question $\boldsymbol{q}_j$ given $\boldsymbol{R}$, and concatenate all answers in order as the final model card. To leverage the LLM's strengths in responding effectively to varied questions, we assign specific roles to the LLM tailored to different questions, and outline its expected areas of expertise. Pre-defined roles include project organizer, sociotechnical practical ethicist, and developer, as shown in Table 1 and Appendix A, according to Raw et al. (2022). See Appendix F for LLM inference details.

6 Evaluation Setup

We evaluate CARDGEN on various standard as well as state-of-the-art metrics to measure the faithfulness, relevance, and other aspects of the generation quality. Additionally, we also incorporate human evaluation for the pipeline to address three key challenges that can't be solved by automatic metrics: First, there is an absence of ground truth labels of generated model cards by CARDGEN. To mitigate this, we have to develop specific manual evaluations to assess performance. Second, current model cards created by human developers are often incomplete and deviate from the recommended template provided by HF. Third, the LLM generated model card is typically long with over 4000 words, and brings challenges to both open-source standard evaluations with limited context size and costly GPT-based metrics.

Standard Metrics We follow Honovich et al. (2022) and use ROUGE (Lin, 2004), BERTScore (Zhang et al., 2019), BARTScore (Yuan et al., 2021), and NLI-finetuned models (Williams et al., 2018; MacCartney and Manning, 2008) to measure the factual consistency of retrieved chunks set $\boldsymbol{R}$ and the generated answer $\boldsymbol{A}$. Due to the large size of retrieved texts, we use `deberta-v3-base` as the base model for BERTScore, and use `nli-deberta-v3-large` as the NLI-finetuned model scorer (Reimers and Gurevych, 2019a; He et al., 2021). More details in Appendix H.

Metric	Input	Description
Factual consistency	$\boldsymbol{R}, \boldsymbol{A}$	How much the generated answer is supported by retrieved contexts.
Faithfulness	$\boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{A}$	How much the statements created from the question-answer pair are supported by the retrieved context.
Answer relevance	$\boldsymbol{Q}, \boldsymbol{A}$	relevance score of the answer according to the given question.
Context precision	$\boldsymbol{Q}, \boldsymbol{R}$	How much the given context is useful in answering the question.
Context relevance	$\boldsymbol{Q}, \boldsymbol{R}$	Whether the question can be answered by relevant sentences extracted from the given context.

Table 3: Illustration of the input and description of standard metrics and GPT-based metrics being used.

Metric	Human	GPT3.5	Llama2 70B	Mistral 7B	Llama2 7B
Completeness	2.33	**3.75**	3.60	3.60	3.70
Accuracy	**4.53**	3.31	3.28	3.00	2.97
Objectivity	2.30	**4.08**	3.15	3.02	3.83
Understandability	2.15	**4.05**	3.85	3.75	3.17
Reference quality	**4.33**	3.55	3.33	3.20	2.70

Table 4: Human evaluation results on LLM generated and human-generated model cards.

GPT Metrics Following Es et al. (2023), we consider the measurement of faithfulness, answer relevance, context precision, and context relevance using GPT4. Table 3 provides a description of these metrics. As different combinations of inputs are taken into consideration, these metrics are necessary supplements to standard metrics. Full prompt details are explained in Appendix H.

Human Evaluation Metrics Putting together LLM generated cards with the human-generated cards as a sample, we devise the following manual evaluation metrics: completeness, accuracy, objectivity, understandability, and reference quality. We design a simple Gradio annotation interface (Abid et al., 2019), and more details are in Appendix I.

7 Results

7.1 Performance Summary

Our human evaluation results are shown in Table 4 and automatic evaluation results are shown in Tables 5 and 6 for model cards. The only difference for the data card generation pipeline is the substitution with data card question templates. Since this is a new text generation task, we provide no baseline results.Therefore, we mainly answer two questions below:

Are our generated model cards better than human-generated ones? We conduct a random sampling of 50 model cards from the test set and compute the average metric scores across all the annotated samples, as shown in Table 4. GPT3.5 demonstrates superior performance over other LLMs and human-generated content in terms of completeness, objectivity, and understandability. This finding aligns with the observations presented below for Tables 5 and 6.

Conversely, the human-generated model cards received higher scores in accuracy and reference quality. This disparity suggests that all LLMs exhibit some degree of hallucination for factual content and reference links in their generation. It is important to note that the human-generated model cards' incompleteness precludes a direct comparison of human evaluation metrics with the metrics used in Tables 5 and 6. Moreover, the insights derived from Table 4 are not obtainable through automatic metrics. We thus conclude that human evaluation metrics are indispensable components of our overall evaluation framework.

How does GPT3.5 perform compared with open sourced LLMs? From Table 5, we can't observe a uniform trend for factual consistency across all sub-tasks. GPT3.5 outperforms open-sourced LLMs on "Uses" and "Bias" question sets in 3 over 4 standard metrics, while Llama2 70b generates more factual consistent answers on other sub-tasks according to ROUGE-L and BERTScore.

According to Table 6, GPT3.5 beats other LLMs on faithfulness and answer relevance across nearly all sub-tasks, and shows its strong instruction-following capabilities for question-answering. However, we have an interesting observation that

Metric	Model	Summary	Model details	Uses	Bias	Training details	Evaluation	More info
ROUGE-L	GPT3.5	9.90	10.70	**16.51**	**20.21**	14.46	15.75	10.73
	Llama2 70b chat	**12.71**	**14.35**	12.85	17.20	**18.74**	**18.03**	**16.21**
	Mistral 7b inst	12.19	11.01	13.02	15.07	16.79	16.23	9.47
	Llama2 7b chat	11.91	12.84	13.89	15.85	14.63	16.21	13.61
BERTScore	GPT3.5	54.86	53.17	**58.62**	**59.29**	56.61	57.42	52.47
	Llama2 70b chat	**57.21**	**56.15**	53.97	56.55	**59.69**	**59.46**	**56.99**
	Mistral 7b inst	55.69	52.80	54.12	53.76	57.10	57.63	49.12
	Llama2 7b chat	55.76	54.51	53.93	55.48	56.30	57.13	54.72
BARTScore	GPT3.5	**17.09**	9.58	2.04	3.52	5.75	6.65	**9.10**
	Llama2 70b chat	14.17	5.41	1.45	3.10	5.30	4.60	5.91
	Mistral 7b inst	16.52	**9.65**	2.00	3.55	**7.00**	**8.75**	8.31
	Llama2 7b chat	14.04	3.49	**2.11**	**3.61**	4.70	3.68	4.01
NLI	GPT3.5	**65.14**	49.83	**57.54**	**62.41**	**59.14**	60.14	56.80
	Llama2 70b chat	56.46	**51.70**	55.22	58.42	57.70	62.04	**59.74**
	Mistral 7b inst	58.67	50.36	54.25	54.59	59.06	58.91	55.17
	Llama2 7b chat	56.46	50.19	54.31	57.23	57.82	**62.11**	56.44

Table 5: Factual consistency evaluation results per section on our retrieve-and-generate pipeline using ROUGE-L, BERTScore, BARTScore, and NLI pretrained scorers.

Metric	Model	Summary	Description	Direct use	Bias, risks, limitation	Results summary
Faithfulness	**GPT3.5**	71.23	**83.21**	**48.71**	**55.17**	**82.99**
	Llama2 70b chat	70.03	76.39	43.20	32.14	63.87
	Mistral 7b inst	**76.75**	75.03	38.28	41.77	73.61
	Llama2 7b chat	72.41	71.35	48.43	44.23	65.56
Answer relevance	**GPT3.5**	**91.18**	**93.26**	90.70	**93.75**	**93.24**
	Llama2 70b chat	90.76	92.27	91.25	92.23	91.63
	Mistral 7b inst	90.46	91.77	90.36	91.56	90.43
	Llama2 7b chat	90.44	90.95	**92.55**	92.69	92.81
Context precision	GPT3.5	29.07	51.80	25.71	18.77	37.88
	Llama2 70b chat	21.05	50.00	25.35	20.03	40.82
	Mistral 7b inst	31.10	**52.22**	**28.45**	**21.36**	**44.45**
	Llama2 7b chat	**32.46**	50.79	25.52	14.27	40.04
Context relevance	**GPT3.5**	13.27	**51.03**	**29.82**	**18.97**	**26.44**
	Llama2 70b chat	13.32	49.62	27.22	18.37	24.31
	Mistral 7b inst	13.22	47.05	28.40	18.75	23.52
	Llama2 7b chat	**13.87**	50.78	28.07	17.57	26.23

Table 6: GPT4 evaluation results on five most important questions based on faithfulness (Faith), answer relevance (AR), context precision (CP), and context relevance (CR).

though GPT3.5 has higher context relevance scores, it is outperformed by Mistral 7B on context precision. A higher context relevance indicates that the question can be better answered from the given context, while a lower context precision means that the context may contain other unnecessary information for answering the question. The discrepancy between results by these two metrics suggests that retrieved texts from the GPT CardGen pipeline are more informative but less concise. Additionally, since we use LLM generated pseudo answers as queries for similar paragraphs, pseudo answers with more possibly unrelated contents will lead to more irrelevant chunks from retrieval. Along with the illustration in Figure 9, we draw the conclusion that GPT3.5 generates pseudo answers with potentially more unrelated details.

7.2 Ablation Study

To evaluate the significance of CardGen's components, we conducted the following ablation studies and explored model architecture variations: (1) Remove the pseudo answer chain and use original questions for embedding similarity matching. (2) Vary the final generation chain only with different LLMs, and maintain all preceding reasoning chains as generated by GPT3.5. (3) Employ different embedding models for dense retrieval. To manage the expenses associated with OpenAI AI calling, we employ GPT3.5 for subsequent studies. We obtain Krippendorff's α (mean=0.83, std=0.14, min=0.56, max=0.99) for the agreements on Table 6 by GPT4 and GPT3.5 to validate our evaluation model substitution (Castro, 2017).

Metric	Model	Summary	Description	Direct use	Bias, risks, limitation	Results summary
NLI	GPT3.5	65.14(+2.14)	51.53(+0.53)	50.51(+0.51)	64.12(+1.12)	58.50(+0.50)
	w/o pseudo	63.00	51.00	50.00	63.00	58.00
Faith	GPT3.5	81.93(+6.75)	79.30(+4.30)	41.23(+0.62)	46.42(-2.53)	72.66(+1.21)
	w/o pseudo	75.18	75.00	40.61	48.95	71.45
AR	GPT3.5	86.94(+0.06)	89.56(-0.65)	88.95(+0.78)	93.55(+0.40)	95.20(+0.02)
	w/o pseudo	86.88	90.21	88.17	93.15	95.18
CP	GPT3.5	47.53(+7.49)	19.61(+1.01)	13.44(+3.20)	13.03(-0.26)	64.15(+0.24)
	w/o pseudo	40.04	18.60	10.24	13.29	63.91
CR	GPT3.5	11.85(+2.32)	23.24(-2.21)	8.70(+1.19)	4.35(+0.69)	24.04(+5.79)
	w/o pseudo	9.53	25.45	7.51	3.66	18.25
Faith	GPT3.5	81.93(8.09)	79.30(15.31)	41.23(26.62)	46.42(22.14)	72.66(25.16)
	Llama2 70B	73.84	63.99	14.61	24.28	47.50
AR	GPT3.5	86.94(-1.56)	89.56(+0.63)	88.95(6.58)	93.55(9.53)	95.20(7.21)
	Llama2 70B	88.50	88.93	82.37	84.02	87.99

Table 7: GPT3.5 evaluation results on five most important questions for pseudo answer chain ablation in top five rows and generation chain ablation in bottom two rows. For the generation chain ablation, we keep all previous chains unchanged with `GPT-3.5-turbo` as the backbone, and only vary the choice of LLMs for the final generation chain, including `GPT-3.5-turbo` and `Llama2-70B-Chat-HF`.

Pseudo Answer Chain We compare the GPT evaluation scores and factual consistency using NLI of CardGen + GPT3.5 pipeline with or without the pseudo answer chain, as illustrated in Table 7. CardGen with the pseudo answer chain outperforms the other across nearly all important questions and metrics being tested. Our results demonstrate the necessity of the pseudo answer chain in our pipeline. Some lower scores may be because of more unrelated texts from the generated pseudo answers for specific questions.

Generation Chain In bottom two rows of Table 7, we show the comparison results by only substituting GPT3.5 in the generation chain with Llama2 70B based on faithfulness and answer relevance. Context precision and context relevance are the same since retrieved texts remain unchanged. We observe a large drop for the faithfulness score and a moderate drop for the answer relevance score, indicating the stronger instruction following capability of GPT3.5 in the generation stage compared to Llama2 70B.

Embedding Models We compare the embedding model `jina-embeddings-v2-base-en` that we use with two other commonly used sentence transformer models: `all-MiniLM-L6-v2` and `all-mpnet-base-v2` (Günther et al., 2023; Wang et al., 2020a; Reimers and Gurevych, 2019b, 2020). We justify our choice of embedding models in Figure 3, where CardGen with `jina-embeddings-v2-base-en` performs better than others according to all three metrics related to the retrieved texts.

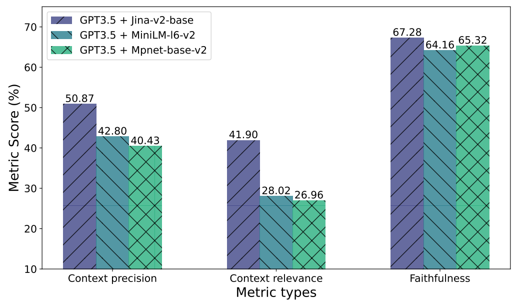

Figure 3: Comparison of three embedding models on context precision, context relevance, and faithfulness.

7.3 LLM Generated Model Card Statistics

Appendix G provides related statistics. Compared with human generated model card statistics in Table 11, LLM generated model cards are longer and more informative.

8 Conclusion

In this study, we introduce a novel task focused on the automatic generation of model cards and data cards. This task is facilitated by the creation of the CardBench dataset, and the development of the CardGen pipeline leveraging state-of-the-art LLMs. The system is designed to assist in the generation of understandable, comprehensive, and consistent models and data cards, thereby providing a valuable contribution to the field of responsible AI.

Limitations

One limitation of our method is that, despite the adoption of the RAG pipeline and explicit instructions for LLMs to adhere closely to the retrieved text, there remains the potential for hallucinations in the generated text. To mitigate this, future work may integrate specific strategies into our CARDGEN pipeline for hallucination reduction by carefully balancing generation speed with quality.

Our current approach employs a single-step generation process and a two-step retrieval process that first infers relevant section contents. Future work could incorporate more advanced chain-of-thought prompting techniques and compare with our CARDGEN pipeline. For complex questions requiring multistep reasoning, after decomposed into manageable sub-questions, we can address each sub-question through multiple reasoning steps, as suggested by recent research (Yao et al., 2022; Khot et al., 2022; Press et al., 2022; He et al., 2022). Additionally, an iterative retrieval-generation collaborative framework can also be used to refine responses in each iteration based on newly retrieved contexts, following recent advancements in iterative retrieval and generation frameworks for complex tasks (Shao et al., 2023; Feng et al., 2023).

Ethical Considerations

This work aims to provide insights about the automatic generation of model cards and data cards. Such an endeavor is instrumental in promoting accountability and traceability among developers as they document their models. The dataset for this research was collected using public REST APIs from HF Hub, Arxiv, and GitHub. We ensured that only open-source model cards, data cards, and their associated source documents were collected, strictly adhering to the stipulations of their respective licenses for research purposes, so there were no user privacy concerns in the dataset. Our dataset and method should only be used for research purpose.

References

Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. 2019. Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv preprint arXiv:1906.02569*.

David Adkins, Bilal Alsallakh, Adeel Cheema, Narine Kokhlikyan, Emily McReynolds, Pushkar Mishra, Chavez Procope, Jeremy Sawruk, Erin Wang, and Polina Zvyagina. 2022. Prescriptive and descriptive approaches to machine-learning transparency. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–9.

Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilović, et al. 2019. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. *arXiv preprint arXiv:1909.03012*.

Solon Barocas and Andrew D. Selbst. 2016. Big data's disparate impact. *California Law Review*, 104:671.

Mario Barrantes, Benedikt Herudek, and Richard Wang. 2020. Adversarial nli for factual correctness in text summarisation models. *arXiv preprint arXiv:2005.11739*.

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Emily M. Bender, Batya Friedman, and Angelina McMillan-Major. 2021. Data statements for nlp: Towards best practices.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Qian Cao and Deyi Xiong. 2018. Encoding gated translation memory into neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3042–3047, Brussels, Belgium. Association for Computational Linguistics.

Santiago Castro. 2017. Fast Krippendorff: Fast computation of Krippendorff's alpha agreement measure. `https://github.com/pln-fing-udelar/fast-krippendorff`.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2023. Complex claim verification with evidence retrieved in the wild. *arXiv preprint arXiv:2305.11859*.

Roxana Daneshjou, Mary P. Smith, Mary D. Sun, Veronica Rotemberg, and James Zou. 2021. Lack of Transparency and Potential Bias in Artificial Intelligence Data Sets and Algorithms: A Scoping Review. *JAMA Dermatology*, 157(11):1362–1369.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mark Díaz, Ian Kivlichan, Rachel Rosen, Dylan Baker, Razvan Amironesei, Vinodkumar Prabhakaran, and Emily Denton. 2022. Crowdworksheets: Accounting for individual and collective identities underlying crowdsourced dataset annotation. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2342–2351.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Zhangyin Feng, Xiaocheng Feng, Dezhi Zhao, Maojin Yang, and Bing Qin. 2023. Retrieval-generation synergy augmented large language models. *arXiv preprint arXiv:2310.05149*.

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

Boris A. Galitsky. 2023. Truth-o-meter: Collaborating with llm in fighting its hallucinations. *Preprints*.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. *arXiv preprint arXiv:2104.08253*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Ross Gruetzemacher and Jess Whittlestone. 2022. The transformative potential of artificial intelligence. *Futures*, 135:102884.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents.

Hangfeng He, Hongming Zhang, and Dan Roth. 2022. Rethinking with retrieval: Faithful large language model inference. *arXiv preprint arXiv:2301.00303*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. 2020. The dataset nutrition label. *Data Protection and Privacy*, 12(12):1.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. True: Re-evaluating factual consistency evaluation. *arXiv preprint arXiv:2204.04991*.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7856–7870, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Ben Hutchinson, Andrew Smart, Alex Hanna, Emily Denton, Christina Greer, Oddur Kjartansson, Parker Barnes, and Margaret Mitchell. 2021. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 560–575.

Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. 2021. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 624–635.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Philippe Laban, Wojciech Kryściński, Divyansh Agarwal, Alexander R Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. Llms as factual reasoners: Insights from existing benchmarks and beyond. *arXiv preprint arXiv:2305.14540*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110*.

Weixin Liang, Girmaw Abebe Tadesse, Daniel Ho, L Fei-Fei, Matei Zaharia, Ce Zhang, and James Zou. 2022. Advances, challenges and opportunities in creating data for trustworthy ai. *Nature Machine Intelligence*, 4(8):669–677.

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

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521–528, Manchester, UK. Coling 2008 Organizing Committee.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Angelina McMillan-Major, Salomey Osei, Juan Diego Rodriguez, Pawan Sasanka Ammanamanchi, Sebastian Gehrmann, and Yacine Jernite. 2021. Reusable templates and guides for documenting datasets and models for natural language processing and generation: A case study of the huggingface and gem data and model cards. *arXiv preprint arXiv:2108.07374*.

Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. 2021. Rethinking search: Making domain experts out of dilettantes. *SIGIR Forum*, 55(1).

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. 2023. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. 2021. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback, 2022. *URL https://arxiv.org/abs/2203.02155*, 13.

Trishan Panch, Heather Mattie, and Rifat Atun. 2019. Artificial intelligence and algorithmic bias: implications for health systems. *Journal of global health*, 9(2).

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

P Jonathon Phillips, Carina A Hahn, Peter C Fontana, David A Broniatowski, and Mark A Przybocki. 2020. Four principles of explainable artificial intelligence. *Gaithersburg, Maryland*, 18.

Malte Pietsch, Soni Tanay, Chan Branden, Möller Timo, and Kostić Bogdan. 2020. Deepset-ai/haystack.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Mahima Pushkarna, Andrew Zaldivar, and Oddur Kjartansson. 2022. Data cards: Purposeful and transparent dataset documentation for responsible ai. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1776–1826.

Nathan Raw, Adrin Jalali, and Sugato Ray. 2022. [link].

Nils Reimers and Iryna Gurevych. 2019a. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Nils Reimers and Iryna Gurevych. 2019b. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389.

Sashank Santhanam, Behnam Hedayatnia, Spandana Gella, Aishwarya Padmakumar, Seokhwan Kim, Yang Liu, and Dilek Hakkani-Tur. 2021. Rome was built in 1776: A case study on factual correctness in knowledge-grounded response generation. *arXiv preprint arXiv:2110.05456*.

Christin Seifert, Stefanie Scherzinger, and Lena Wiese. 2019. Towards generating consumer labels for machine learning models. In *2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)*, pages 173–179. IEEE.

Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. 2022. Compute trends across three eras of machine learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *arXiv preprint arXiv:2305.15294*.

Hong Shen, Wesley H. Deng, Aditi Chattopadhyay, Zhiwei Steven Wu, Xu Wang, and Haiyi Zhu. 2021. Value cards: An educational toolkit for teaching social impacts of machine learning through deliberation. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 850–861, New York, NY, USA. Association for Computing Machinery.

Karan Shukla, Suzen Fylke, Hannes Hapke, Kalvin Leung, et al. 2021. Model card toolkit.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Pablo Villalobos, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, Anson Ho, and Marius Hobbhahn. 2022. Machine learning model sizes and the parameter gap. *arXiv preprint arXiv:2207.02852*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020a. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020b. Towards faithful neural table-to-text generation with content-matching constraints. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1072–1086, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Yue Yu, Chenyan Xiong, Si Sun, Chao Zhang, and Arnold Overwijk. 2022. COCO-DR: Combating distribution shift in zero-shot dense retrieval with contrastive and distributionally robust learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1462–1479, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

A Question Templates

Tables 8 and 9 shows full question templates of model cards and data cards. We have 31 questions in total for generating model cards, and 21 questions for generating data cards. We create these questions based on the template provided by HF.[8] and include necessary requirements

B Dataset Collection Details

For the model card evaluation set selection, we select all 350 examples that are rewritten by the HF team with their unique disclaimers, as shown in Figure 4.

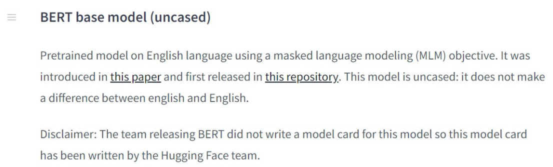

Figure 4: `bert-base-uncased` (Devlin et al., 2018) as a current model card example with a unique disclaimer sentence, indicating a modification by the HF team.

C Dataset Annotation Details

Human Annotation Guidelines To evaluate paper links and direct GitHub links on the model card evaluation set, we require the annotators to go through each current model card and provide all possible paper links and GitHub links to annotators. They are asked to select the direct paper link and GitHub link from all candidate links, by looking at their positions of occurrences in the model card example. If no direct links of either sources can be determined, they need to label this model card as "Invalid".

GPT Annotation Details We show our two-shot prompts for asking `GPT-3.5-turbo` to select direct paper links in Figure 5. Direct GitHub link selection is prompted similarly.

[8] `https://github.com/huggingface/huggingface_hub/tree/main/src/huggingface_hub/templates`

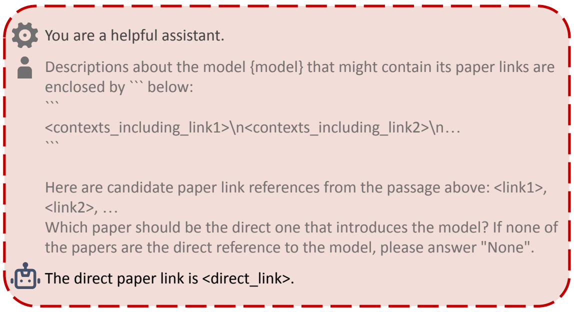

Figure 5: Prompts for calling GPT3.5 to select direct paper links. We prepend one positive example and one negative example to the message list to improve its inference quality.

LLM	# words	# sentences	# links
GPT3.5	4023.88	215.17	4.18
Llama2 70B Chat	6210.32	323.56	4.55
Llama2 7B Chat	5548.50	302.73	1.44
Mistral 7B Inst	4126.07	202.16	2.65

Table 10: Statistics about whole generated model cards

D Dataset Analysis

We provide the number of card examples with direct paper links in their human-generated cards, with direct GitHub repository links, and with both links in Table 11. We also provide additional figures about the dataset task taxonomy in Figure 6. The taxonomy is obtained using the REST API of HF Hub.

	Split	Measure	# W papers	# W repos	# W both
ModelCard	all	# samples	5689	4829	2485
		# words	1064	948	1134
	test	# samples	344	299	294
		# words	668	710	711
DataCard	all	# samples	660	533	328
		# words	1394	1104	1416
	test	# samples	86	71	50
		# words	1003	1290	1155

Table 11: Statistics for crawled model cards and data cards, including the number of examples with direct paper links or direct github links or both, and the average number of words in each category.

E Retriever Details

We use FAISS as our embedding store database (Johnson et al., 2019). We fix the chunk size as 512 and the chunk overlap as 64. After retrieving relevant sections, we choose to obtain 8 chunks from these sections, together with 4 other chunks from other sections to reduce the bias propagation.

Question	Role	Prompt
Summary	Project organizer	Provide a 1-2 sentence summary of what the model is.
Description	Project organizer	Provide basic details about the model. This includes the model architecture, training procedures, parameters, and important disclaimers.
Funded by	Project organizer	List the people or organizations that fund this project of the model.
Shared by	Developer	Who are the contributors that made the model available online as a GitHub repo?
Model type	Project organizer	Summarize the type of the model in terms of the training method, machine learning type, and modality in one sentence.
Language	Project organizer	Summarize what natural human language the model uses or processes in one sentence.
License	Project organizer	Provide the name and link to the license being used for the model.
Finetuned from	Project organizer	If the model is fine-tuned from another model, provide the name and link to that base model.
Demo sources	Project organizer	Provide the link to the demo of the model.
Direct use	Project organizer	Explain how the model can be used without fine-tuning, post-processing, or plugging into a pipeline. Provide a code snippet if necessary
Downstream use	Project organizer	Explain how this model can be used when fine-tuned for a task or when plugged into a larger ecosystem or app. Provide a code snippet if necessary
Out of scope use	Sociotechnic	How the model may foreseeably be misused and address what users ought not do with the model.
Bias risks limitations	Sociotechnic	What are the known or foreseeable issues stemming from this model? These include foreseeable harms, misunderstandings, and technical and sociotechnical limitations.
Bias recommendations	Sociotechnic	What are recommendations with respect to the foreseeable issues about the model?
Training data	Developer	Write 1-2 sentences on what the training data of the model is. Links to documentation related to data pre-processing or additional filtering may go here as well as in More Information.
Preprocessing	Developer	Provide detail tokenization, resizing/rewriting (depending on the modality), etc. about the preprocessing for the data of the model.
Training regime	Developer	Provide detail training hyperparameters when training the model.
Speeds sizes times	Developer	Provide detail throughput, start or end time, checkpoint sizes, etc. about the model.
Testing data	Developer	Provide benchmarks or datasets that the model evaluates on.
Testing factors	Sociotechnic	What are the foreseeable characteristics that will influence how the model behaves? This includes domain and context, as well as population subgroups. Evaluation should ideally be disaggregated across factors in order to uncover disparities in performance.
Testing metrics	Developer	What metrics will be used for evaluation in light of tradeoffs between different errors about the model?
Results	Developer	Provide evaluation results of the model based on the Factors and Metrics.
Results summary	Developer	Summarize the evaluation results about the model.
Model examination	Developer	This is an experimental section some developers are beginning to add, where work on explainability/interpretability may go about the model.
Hardware	Developer	Provide the hardware type that the model is trained on.
Software	Developer	Provide the software type that the model is trained on.
Hours used	Developer	Provide the amount of time used to train the model.
Cloud provider	Developer	Provide the cloud provider that the model is trained on.
Co2 emitted	Developer	Provide the amount of carbon emitted when training the model.
Model specs	Developer	Provide the model architecture and objective about the model.
Compute infrastructure	Developer	Provide the compute infrastructure about the model.

Table 8: Template of the all questions necessary for generating a whole model card.

F Generator Details

Open-sourced LLMs are inferenced through `vllm` Kwon et al. (2023). `Llama2-70B-Chat-HF` is run on 4 A6000s. Two 7B models are run on 1 A6000. We fix temperature to 0 to ensure a stable generation quality. We show our prompt description of different roles in Table 12, and the generation prompt in Figure 10.

G LLM Generated Model Card Statistics

H Metric Details

For standard metrics, we use the list of retrieved texts together with the generated answer as inputs. We normalize all these scores to be in the [0,1] range. Since the output of `nli-deberta-v3-large` is in {"contradiction", "entailment", "neutral"}, we map these outputs to {0, 0.5, 1}, respectively to maintain a percent-

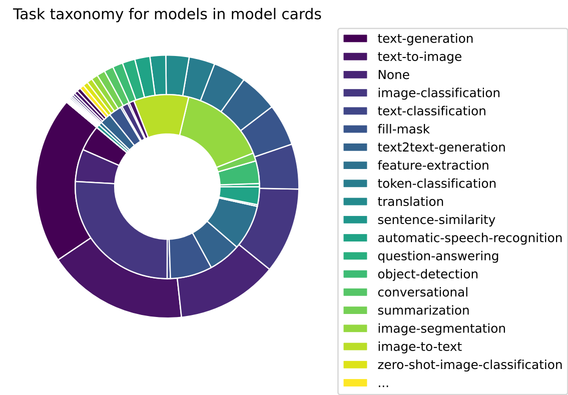

Figure 6: The task taxonomy of models in the model cards dataset, with the inner circle as the test set, and the outer circle as the whole set.

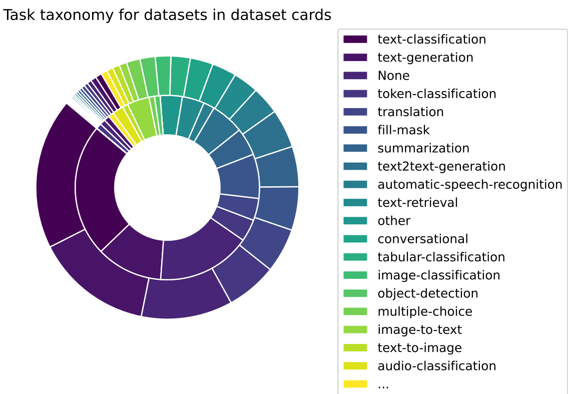

Figure 7: The task taxonomy of datasets in the data cards dataset, with the inner circle as the test set, and the outer circle as the whole set.

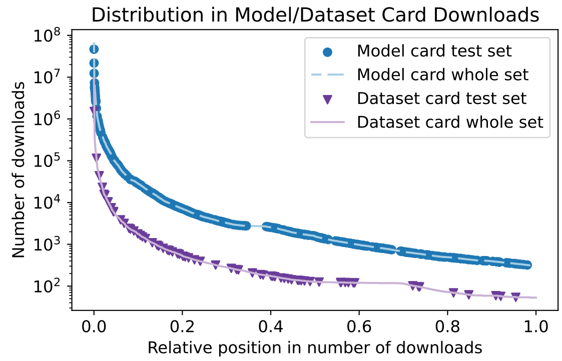

Figure 8: Distribution of the amount of downloads for the whole dataset and the test set. Test set examples distribute quite uniformly.

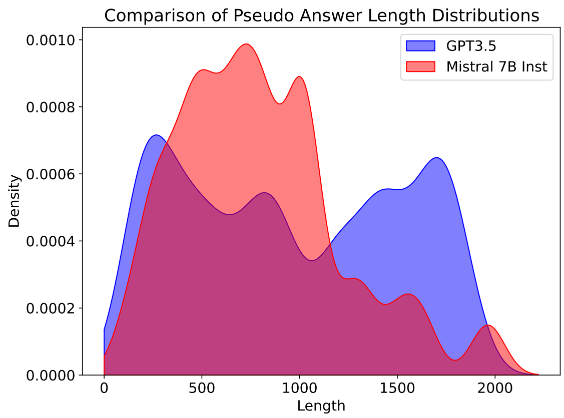

Figure 9: Distribution comparison of pseudo answer length generated by GPT3.5 and Mistral 7B Instruct.

age scale. We use the implementation of ROUGE score by HF. We use official implementations for BERTScore and BARTScore.

For GPT metrics, we use `GPT-4-1106-preview` as evaluators for the main results, and use `GPT-3.5-turbo` for ablation studies.

I Human Annotation Details

We give two annotators the same set of examples each with four model cards generated by LLMs and one written by human. We calculate the Krippendorff's α among the results of two annotators, and got mean=0.76 and std=0.13 for the agreement level. We report averaged ranking scores in Table 4. Note that we don't have direct comparison across human evaluation metrics vs. automatic metrics, since our human metrics evaluate on a whole model card, while automatic metrics take each $(\boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{A})$ tuple for evaluation and they have different scales. We need to implement human metrics in this way to supplement the limited scope of automatic metrics' focus. The annotation interface is shown in Figure 11.

J Pseudo Answer Analyses

We show the distribution of pseudo answer length generated by GPT3.5 and Mistral 7B Instruct in Figure 9.

Statistics about LLM generated model cards are shown in Tables 10 and 13 to 16.

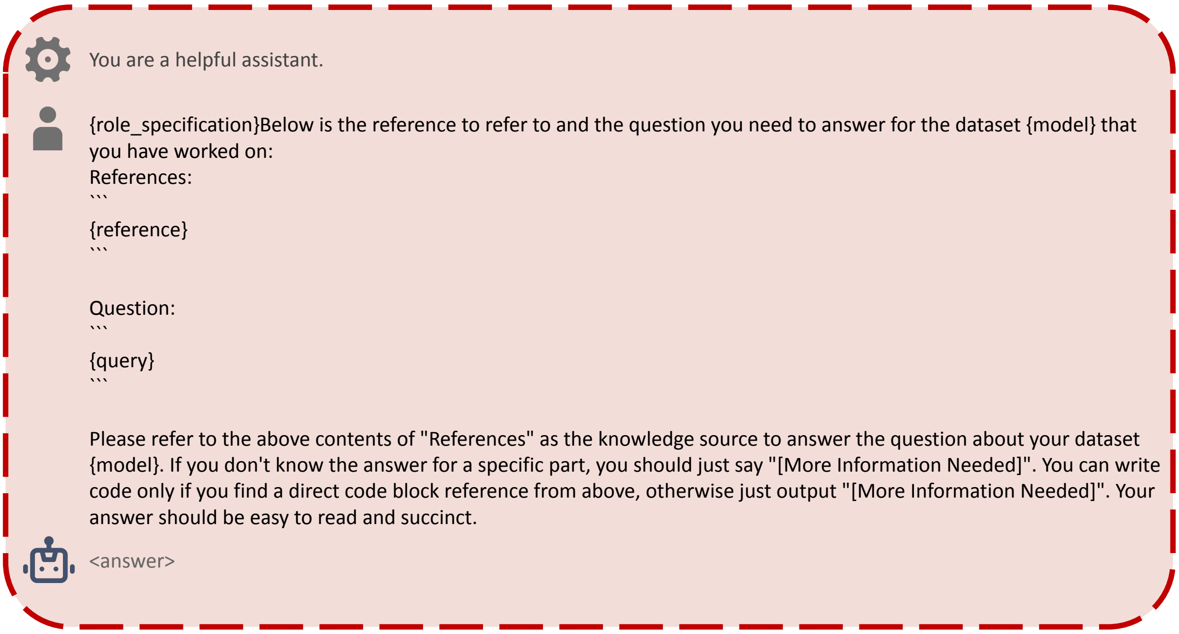

Figure 10: Our generation prompt templates.

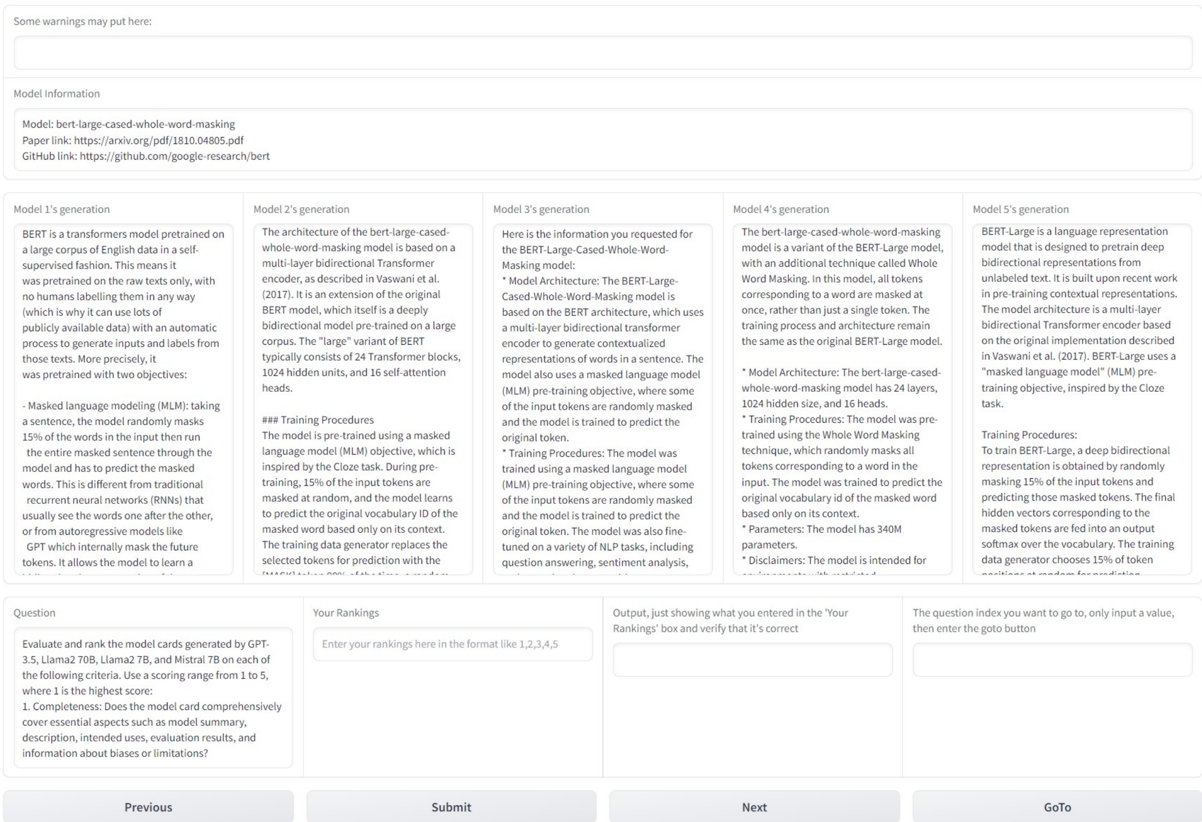

Figure 11: The human annotation interface built by gradio with an example of model `bert-large-cased-whole-word-masking` (Abid et al., 2019; Devlin et al., 2018). The information that a model card is written by whom is hidden, and orders of five model cards shown at each time are randomly shuffled to avoid positional bias.

Question	Role	Prompt
Description	Data manager	Provide the homepage link for the dataset, just give me a link please.
Leaderboard	Data manager	Provide the Leaderboard link for the dataset.
Pointofcontact	Data manager	Provide the Point of Contact for the dataset.
Summary	Data manager	Provide basic details about the dataset. Briefly summarize the dataset, its intended use and the supported tasks. Give an overview of how and why the dataset was created. The summary should explicitly describe the domain, topic, or genre covered.
Supported tasks and leaderboards	Data analyst	Describe the tasks and leaderboards supported by the dataset. Include task description, metrics, suggested models, and leaderboard details.
Languages	Data analyst	Provide an overview of the languages represented in the dataset, including details like language type, script, and region. Include BCP-47 codes if available.
Data instances	Data scientist	Provide a JSON-formatted example of a typical instance in the dataset with a brief description. Include a link to more examples if available. Describe any relationships between data points.
Data fields	Data architect	List and describe the fields in the dataset, including their data type, usage in tasks, and attributes like span indices. Mention if the dataset contains example IDs and their inherent meaning.
Data splits	Data manager	Describe the data splits in the dataset. Include details such as the number of splits, any criteria used for splitting the data, differences between the splits, and the sizes of each split. Provide descriptive statistics for the features where appropriate, for example, average sentence length for each split.
Curation rationale	Data manager	What need or purpose motivated the creation of this dataset? Describe the underlying reasons and major choices involved in its assembly. Explain the significance of the dataset in its field and any specific gaps or demands it aims to address.
Source data	Data manager	Describe the source data used for this dataset. Describe the data collection process. Describe any criteria for data selection or filtering. List any key words or search terms used. If possible, include runtime information for the collection process.
Source language producers	Data manager	Clarifying the human or machine origin of the dataset. Avoiding assumptions about the identity or demographics of the data creators. Providing information about the people represented in the data, with references where applicable.
Annotations	Data manager	Describe the annotation process to the dataset. Detail the annotation process and tools used, or note if none were applied. Specify the volume of data annotated.
Annotators	Data manager	Describe the annotator of the dataset. For annotations in the dataset, state their human or machine-generated nature. Describe the creators of the annotations, their selection process, and any self-reported demographic information.
Personal and sensitive information	Data manager	Categorize how identity data, such as gender referencing Larson (2017), is sourced and used in the dataset. Indicate if the data includes sensitive information or can identify individuals. Describe any anonymization methods applied.
Social impact of dataset	Data manager	Explore the dataset's social impacts: its role in advancing technology and enhancing quality of life. Consider negative effects like decision-making opacity and reinforcing biases. Check if it includes low-resource or under-represented languages. Assess its impact on underserved communities.
Discussion of biases	Data manager	When constructing datasets, especially those including text-based content like Wikipedia articles, biases may be present. If there have been analyses to quantify these biases, it's important to summarize these studies and note any measures taken to mitigate the biases.
Other known limitation	Data analyst	Outline and cite any known limitations of the dataset, such as annotation artifacts, in your studies.
Dataset curators	Data manager	List the people involved in collecting the dataset and their affiliations. If known, include information about funding sources for the dataset. This should encompass individuals, organizations, and any collaborative efforts involved in the dataset creation.
Licensing information	Legal advisor	Provide the license and link to the license webpage if available for the dataset.
Contributions	Data manager	Write in 1-2 sentence about the contributers for the dataset. Mention the GitHub username and provide their GitHub profile link. You should follows the format: Thanks to [@github-username](https://github.com/<github-username>) for adding this dataset.

Table 9: Template of the all questions necessary for generating a whole data card.

Card	Role	Description
ModelCard	Developer	who writes the code and runs training
	Sociotechnic	who is skilled at analyzing the interaction of technology and society long-term (this includes lawyers, ethicists, sociologists, or rights advocates)
	Project organizer	who understands the overall scope and reach of the model and can roughly fill out each part of the card, and who serves as a contact person for model card updates
DataCard	Data curator	who collects and organizes the data
	Data analyst	who is skilled at understanding and documenting dataset characteristics and biases
	Data manager	who oversees dataset versioning, availability, and usage guidelines

Table 12: Our prompts for different roles in answering specific questions.

Question	# words	# sentences	# links
Summary	53.91	1.95	0.02
Description	275.47	14.51	0.17
Funded by	78.29	4.25	0.37
Shared by	33.41	1.86	0.36
Model type	46.11	1.51	0.00
Language	30.24	1.10	0.01
License	47.56	2.78	0.53
Finetuned from	93.95	4.81	0.26
Demo sources	76.70	3.83	0.66
Direct use	227.26	8.78	0.34
Downstream use	287.05	10.23	0.17
Out of scope use	305.64	16.50	0.20
Bias risks limitations	305.09	19.07	0.01
Bias recommendations	298.46	18.04	0.04
Training data	61.17	3.14	0.29
Preprocessing	169.67	11.06	0.04
Training regime	110.71	4.82	0.00
Speeds sizes times	170.33	8.41	0.21
Testing data	112.20	7.98	0.01
Testing factors	230.03	13.26	0.01
Testing metrics	64.45	3.67	0.01
Results	137.94	7.69	0.03
Results summary	154.57	9.01	0.04
Model examination	214.29	11.32	0.19
Hardware	24.87	1.73	0.00
Software	64.71	3.50	0.03
Hours used	27.95	2.06	0.01
Cloud provider	26.13	1.82	0.03
Co2 emitted	36.01	2.40	0.01
Model specs	207.91	10.52	0.11
Compute infrastructure	51.80	3.59	0.02

Table 13: GPT3.5 generated model card statistics per question averaged by all samples in the test set.

Question	# words	# sentences	# links
Summary	89.40	3.23	0.05
Description	276.50	13.87	0.04
Funded by	96.10	4.96	0.06
Shared by	108.62	4.53	0.58
Model type	115.77	3.47	0.00
Language	100.23	4.30	0.00
License	94.86	4.74	0.82
Finetuned from	137.65	5.96	1.06
Demo sources	150.54	7.42	0.82
Direct use	247.95	7.45	0.05
Downstream use	256.03	8.11	0.03
Out of scope use	341.98	21.69	0.00
Bias risks limitations	330.94	22.76	0.00
Bias recommendations	333.96	22.13	0.01
Training data	103.41	4.54	0.24
Preprocessing	285.66	18.20	0.03
Training regime	208.14	12.66	0.03
Speeds sizes times	250.69	12.74	0.10
Testing data	144.15	9.00	0.01
Testing factors	293.02	17.23	0.00
Testing metrics	267.89	14.11	0.02
Results	276.72	16.85	0.05
Results summary	230.82	10.94	0.03
Model examination	317.01	17.74	0.04
Hardware	81.48	4.29	0.02
Software	91.29	4.54	0.12
Hours used	172.74	7.52	0.02
Cloud provider	82.82	4.38	0.11
Co2 emitted	220.29	9.14	0.11
Model specs	276.66	12.12	0.04
Compute infrastructure	227.01	12.94	0.05

Table 14: Llama2 70B Chat generated model card statistics per question averaged by all samples in the test set.

Question	# words	# sentences	# links
Summary	71.93	2.61	0.00
Description	187.40	8.93	0.01
Funded by	91.97	6.40	0.05
Shared by	57.94	3.18	0.04
Model type	67.69	2.68	0.00
Language	57.52	1.84	0.00
License	43.05	2.79	0.17
Finetuned from	115.16	5.98	0.30
Demo sources	228.09	12.81	0.51
Direct use	260.14	12.20	0.01
Downstream use	301.56	16.29	0.02
Out of scope use	339.81	20.71	0.00
Bias risks limitations	317.83	19.05	0.00
Bias recommendations	336.44	19.88	0.00
Training data	72.18	3.31	0.00
Preprocessing	228.65	13.34	0.00
Training regime	162.46	7.19	0.01
Speeds sizes times	211.52	10.62	0.02
Testing data	87.29	5.55	0.03
Testing factors	344.08	21.64	0.00
Testing metrics	226.08	14.20	0.00
Results	263.82	16.22	0.03
Results summary	215.33	9.79	0.04
Model examination	264.26	15.67	0.02
Hardware	72.26	3.43	0.04
Software	49.32	2.45	0.00
Hours used	164.28	8.29	0.00
Cloud provider	56.88	2.92	0.04
Co2 emitted	243.23	10.27	0.00
Model specs	204.47	9.90	0.01
Compute infrastructure	205.86	12.61	0.05

Table 15: Llama2 7B Chat generated model card statistics per question averaged by all samples in the test set.

Question	# words	# sentences	# links
Summary	63.61	2.39	0.01
Description	264.11	12.87	0.04
Funded by	31.15	1.89	0.06
Shared by	43.69	2.41	0.12
Model type	56.07	1.70	0.00
Language	21.67	1.09	0.01
License	42.63	2.49	0.36
Finetuned from	65.91	3.47	0.49
Demo sources	141.35	6.48	0.94
Direct use	211.97	6.29	0.09
Downstream use	254.17	7.30	0.04
Out of scope use	225.52	10.20	0.00
Bias risks limitations	274.26	16.36	0.00
Bias recommendations	309.82	18.44	0.00
Training data	85.98	4.01	0.02
Preprocessing	222.67	12.46	0.01
Training regime	179.76	11.08	0.01
Speeds sizes times	192.81	9.40	0.05
Testing data	87.16	4.96	0.02
Testing factors	245.14	11.60	0.01
Testing metrics	137.77	7.12	0.01
Results	210.40	10.50	0.04
Results summary	136.51	6.21	0.09
Model examination	169.52	8.47	0.02
Hardware	21.44	1.39	0.01
Software	23.53	1.47	0.04
Hours used	58.86	2.86	0.01
Cloud provider	18.55	1.32	0.02
Co2 emitted	33.65	2.13	0.00
Model specs	161.47	7.17	0.03
Compute infrastructure	134.92	6.61	0.10

Table 16: Mistral 7B Inst generated model card statistics per question averaged by all samples in the test set.