Automatic Model Card Generation Using an LLM

TAJKIA RAHMAN TOMA, University of Alberta, Canada
BALREET GREWAL, University of Alberta, Canada
COR-PAUL BEZEMER, University of Alberta, Canada

Model cards are structured documents that summarize key information about machine learning models to improve transparency, usability, and accountability. However, they often lack a consistent structure, and many models provide no model cards, making comparison and interpretation difficult. This paper presents two contributions. First, we propose MCTidy, an LLM-based approach that reorganizes existing model cards into a standardized template to improve clarity and comparability. Second, we introduce MCGenie, an LLM-based system that generates model cards directly from model repository data. We apply MCTidy to 48 Hugging Face model cards and evaluate information retention, section alignment, hallucination, and stability. Our findings show high information retention with minimal textual loss, accurate section assignment, rare hallucinations primarily in descriptive sections, and strong stability across runs. We assess MCGenie by generating model cards for the same 48 models and assessing semantic similarity, factual correctness, and sensitivity to input resources. The generated model cards achieved high semantic similarity (mean ≈ 0.9); over half were fully correct, and most remaining errors were minor. Generation quality depended strongly on the availability of supporting resources, particularly associated papers. Overall, our findings demonstrate the potential of LLM-based methods to enable scalable, standardized model card documentation.

CCS Concepts: • **Software and its engineering** → **Documentation**; Automatic programming; • **Applied computing** → *Document preparation.*

Additional Key Words and Phrases: Automation, Model Card, LLM

ACM Reference Format:
Tajkia Rahman Toma, Balreet Grewal, and Cor-Paul Bezemer. 2026. Automatic Model Card Generation Using an LLM. 1, 1 (August 2026), 33 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Machine learning (ML) models are increasingly deployed in high-stakes domains such as healthcare [13, 14], finance [2, 11], and criminal justice [35, 39]. Model documentation plays a crucial role in providing developers and users with essential information about the development, limitations, and appropriate use of these ML models [21, 28]. Effective documentation enhances model transparency [21], which in turn supports accountability [38] and enables users to compare, select, and apply models more effectively [32]. In contrast, poor documentation, or the absence of documentation, can lead to misuse, misunderstanding, and frustration among users who need to understand a model's behavior, requirements, and performance in order to make informed decisions [4].

Authors' Contact Information: Tajkia Rahman Toma, tajkiatoma@ualberta.ca, University of Alberta, Edmonton, Canada; Balreet Grewal, balreet@ualberta.ca, University of Alberta, Edmonton, Canada; Cor-Paul Bezemer, bezemer@ualberta.ca, University of Alberta, Edmonton, Canada.

Manuscript submitted to ACM

Model cards [21] have emerged as a widely endorsed practice for documenting ML models [4, 19, 22, 24]. However, many models either lack documentation entirely or are insufficiently documented [4, 19], increasing the risk of model misuse [4]. Despite its importance, creating and maintaining comprehensive, standardized documentation remains a significant challenge [17, 25]. Writing model documentation from scratch is time-consuming, cognitively demanding, and often not prioritized by model developers, especially in fast-paced or resource-constrained environments [4, 17]. Automating the generation of model documentation not only ensures that more model providers provide essential information about the models, but also encourages broader adoption of standardized documentation practices.

Many recent studies have proposed solutions to support automatic model documentation generation [4, 8, 9, 28, 36]. These approaches, however, operate in constrained settings, such as relying on data stored in ML Metadata [12], descriptions available in computational notebooks, or focusing on alternative documentation formats. In contrast, we propose a solution that leverages resources that are often naturally generated during the model development process, such as model repository files or academic papers, reducing the need for additional structured inputs or specialized environments. By leveraging a large language model (LLM) to produce structured documentation based on such available resources, we can enhance the coverage, consistency, and accessibility of model information at scale.

Our study makes two contributions that build on each other. As our first contribution, we introduce MCTidy, an approach for automatically reorganizing the content of existing model cards using an LLM to align them with a standard model card template. The goal is to improve the structure and readability of key information and enhance consistency across model documentation, thereby facilitating broader adoption of best practices and enabling more effective comparison and evaluation of models. Using off-the-shelf LLMs such as Gemini 2.0 Flash Thinking, MCTidy allows model authors to reorganize their documentation easily, without the need for custom infrastructure or extensive manual edits. We reorganized the content of 48 existing Hugging Face model cards and conducted a systematic evaluation of MCTidy along three research questions:

- **RQ1.1: How correct are the reorganizations performed by MCTidy?** Due to the generative nature of LLMs, evaluating content retention after the reorganization and correct information placement are essential to ensure the integrity of reorganized model cards. MCTidy retains most content during the reorganization: a median of only 4.5% of information checklist items derived from the original model cards were missing in the reorganized versions, with minimal textual omission. MCTidy also placed content correctly in nearly all cases, with only 1.9% of the (sub)sections containing misplaced content. These findings suggest that MCTidy performs well in both retaining information and assigning it to appropriate sections, making it a reliable tool for automated model card standardization.
- **RQ1.2: How much hallucinated content is introduced by MCTidy?** Since hallucinations and misinterpretations are known limitations of LLMs, it is essential to assess their impact on content reorganization to determine the level of trust we can place in MCTidy. The overall information accuracy remained high, with only minimal textual changes required to correct the hallucinations and misinterpretations. The hallucinations and misinterpretations were concentrated in specific (sub)sections, suggesting that minimal human oversight focused on these sections is sufficient to catch occasional errors and ensure the overall information accuracy of the reorganized model cards.
- **RQ1.3: How consistent is MCTidy across multiple runs?** Given the inherently non-deterministic nature of LLMs, we evaluate the consistency of MCTidy's outputs to assess the stability of its content reorganization across multiple runs. The results demonstrate high semantic consistency, with the median average semantic similarity

scores across three runs reaching 0.97. Additionally, 87.5% of (sub)sections achieved semantic similarity scores of 0.90 or higher, further supporting MCTidy's suitability for dependable large-scale documentation restructuring.

We released both MCTidy and the 48 reorganized, manually verified model cards as part of our replication package [34]. These verified reorganized model cards not only serve as a benchmark for future research on automated model card generation but also enable scalable, section-by-section analysis of current documentation practices.

Building on this benchmark, our second contribution explores how LLMs can assist in automatically generating a model card from model repository files. While reorganization ensures structural consistency for existing model cards, many models still lack documentation altogether. In such cases, generation becomes essential. LLMs, with their ability to process and produce text reminiscent of human writing, could potentially automate and streamline the creation of comprehensive, standardized documentation. This would reduce the burden on developers, improve accessibility for users, and encourage more effective use of ML models. To evaluate this potential, we use our reorganized and validated dataset as the ground truth for assessing our approach for automatically generating model cards (MCGenie), focusing on the following research questions:

- **RQ2.1: How semantically similar are the generated contents to the original model cards?** We examined how semantically similar the generated model cards are to the original ones, in order to assess MCGenie's ability to accurately capture and convey the key information from repository files. The generated model cards showed high overall semantic similarity (mean = 0.9), indicating strong information retention. However, sections demanding interpretation or reasoning (such as `Caveats and Recommendations`) had lower similarity, suggesting that while MCGenie effectively preserves factual content, it is less consistent in replicating interpretive or context-dependent text.
- **RQ2.2: How correct is the information in the generated model cards?** We assessed whether the information generated by MCGenie is factually correct, as LLMs may introduce speculative or unsupported details. Over half (54.17%) of the generated model cards were entirely correct, and from the remainder, a median generated model card contained only one inaccurate (sub)section. These inaccuracies mainly stemmed from citation-content mismatches or speculative reasoning, suggesting that while MCGenie reliably reproduces factual content, improving source alignment and grounding could further enhance its factual accuracy.
- **RQ2.3: Which input data is the most important for generating model cards?** We examined how variations in input data influenced the generated model cards to understand which repository resources most strongly affect content quality. Repository papers had the greatest impact while configuration or tokenizer files had minimal effect. This implies that papers serve as the primary, information-rich source for MCGenie, whereas other technical files provide supplementary but non-essential details.

The remainder of the paper is structured as follows. Section 2 discusses our study setup. Sections 3 and 4 discuss the results and implications of our study on MCTidy and MCGenie, respectively. Section 5 discusses the threats to the validity of this work, and Section 6 summarizes the related work. Section 7 concludes our work.

2 Study Setup

We reorganize the content of model cards according to a standard template by using an off-the-shelf LLM. We chose the model card template proposed by Mitchell et al. [21] as a standard due to its large adoption in recent studies [4, 19, 22, 24, 33]. We provided the LLM with both the original model card and the standard template, along with instructions on how to reorganize the content. The template includes descriptions for each section and subsection to guide the LLM

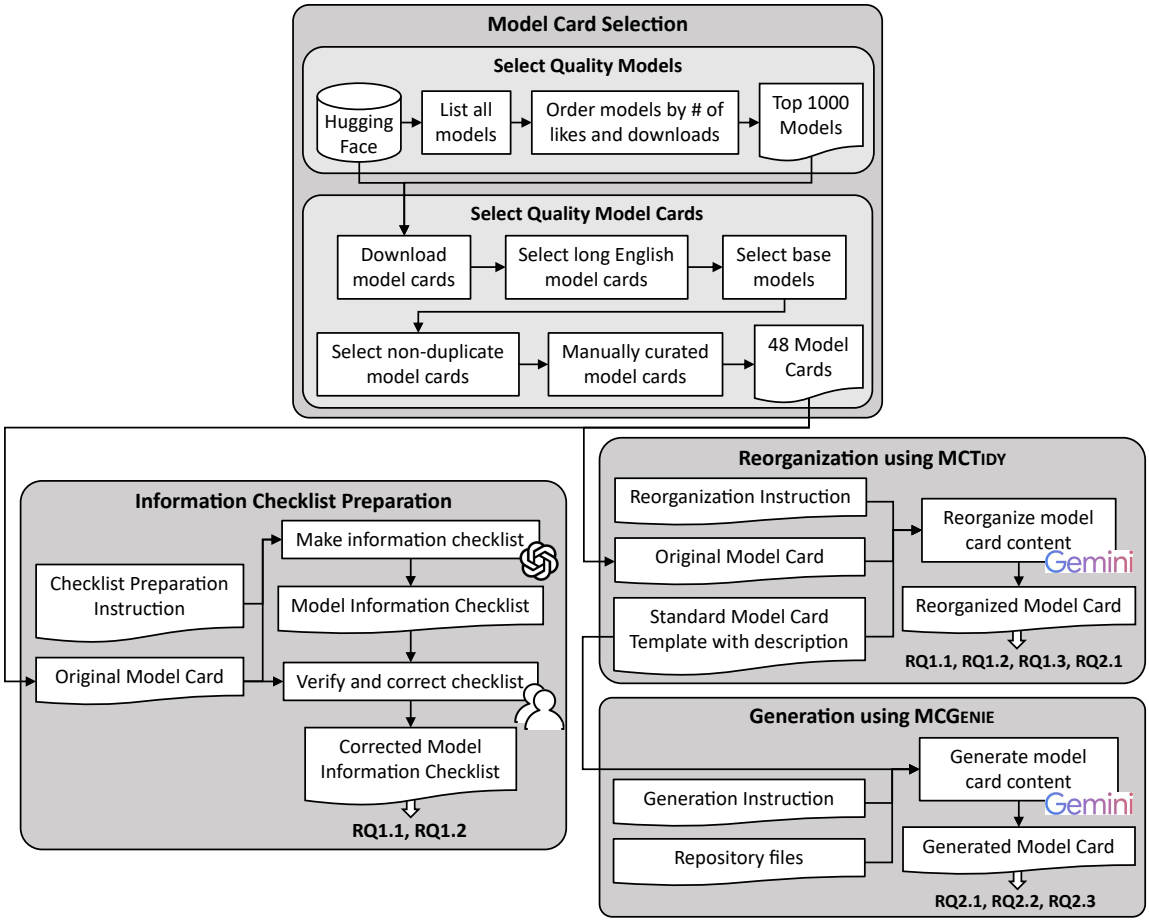

Fig. 1. Overview of our study methodology

during reorganization. The descriptions are based on the framework outlined by Mitchell et al. [21]. We further refined the descriptions using insights from Toma et al. [33], and included the two additional sections proposed in their study: "`How to Use`" and "`Memory and Hardware Requirements`". The description of the sections of the model card template is present in section 7.

Similar to model card reorganization, we generate a model card that follows the same standard template used for reorganization, using an off-the-shelf LLM. However, instead of reorganizing an existing model card, we provided the model with general documentation and resources about the model, such as the associated paper of the model, the tokenizer information saved in JSON file format, configuration files, etc. For the section and subsection descriptions in the template, we reused the descriptions defined during the model card reorganization. We instructed the LLM to structure the information from the provided resources into the appropriate sections and subsections of a model card in accordance with the given template.

In this section, we describe the collection of quality model cards from Hugging Face used for two purposes: reorganizing existing model cards to conform to a standardized template (MCTidy) to evaluate the effectiveness of the

reorganization process, and generating new model cards from model repository data (MCGenie) to assess the quality of the generation. We then explain the data preparation steps undertaken to facilitate the evaluation of MCTidy. Finally, we detail the workflows of both MCTidy and MCGenie. Figure 1 provides an overview of the overall process, which is elaborated upon in the remainder of this section.

2.1 Model Card Selection

Although our reorganization approach can be applied to any model card, we select quality models and their corresponding model cards to effectively evaluate whether the reorganization process can handle long and content-rich documentation. These same model cards were also used to evaluate the generation approach, allowing us to compare the generated model cards with human-written ones in terms of resemblance.

Select Quality Models: We used the Hugging Face Hub API[1] to collect a list of 944,178 models from the Hugging Face Hub. We then sorted them by the number of likes (indicating overall popularity) and by downloads in the last month (showing current popularity). Then we selected the top 1,000 models, assuming that their model cards are well-maintained.

Select Quality Model Cards: We downloaded the model cards of the top 1,000 models for further processing. Since Hugging Face renders `README.md` files as model cards[2], we retrieved the `README.md` files from the repositories using the `huggingface_hub` library[3]. We found 993 repositories with model cards. Then, we filtered them for quality using the following selection criteria:

- **English model cards:** To ensure consistent understanding of the content, we chose to include model cards written only in English, the most commonly used language. We filtered out non-English model cards using `xlm-roberta-base-language-detection` [20], leaving us with 926 model cards.
- **Long model cards:** We selected model cards that are sufficiently detailed, retaining those with at least 1,000 words of descriptive content, excluding code blocks from the word count. Since code blocks focus on how to use the model or output of the model, rather than describing the model, a large code block can make a model card long without adding enough details about the model itself (e.g., the model card of `MEETING_SUMMARY`[4]). This step resulted in 231 model cards.
- **Model cards of base models:** Model cards for derivative models (adapters, fine-tunes, merges, and quantizations) often extend the base model's card. For example, the model cards of `Llama-2-7b-chat-hf`[5] and `Llama-2-7B-Chat-GGML`[6] show how model cards of quantized models combine content from both the base model and the quantized model. Therefore, to focus on original content, we selected only base models and excluded all model cards of derivative models. Whether a model is a base or derivative is indicated in the model card's YAML metadata. After this filtering step, we were left with 176 model cards of base models.
- **Non-duplicate model cards:** Model cards from the same model family (e.g., fine-tuned from the same base model with different numbers of parameters) are often quite similar (e.g., the model card of `gemma-7b`[7] and

[1]https://huggingface.co/docs/huggingface_hub/package_reference/hf_api#huggingface_hub.HfApi.list_models
[2]https://huggingface.co/docs/hub/en/model-cards#model-card-metadata
[3]https://huggingface.co/docs/huggingface_hub/en/guides/download
[4]https://huggingface.co/knkarthick/MEETING_SUMMARY
[5]https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[6]https://huggingface.co/TheBloke/Llama-2-7B-Chat-GGML
[7]https://huggingface.co/google/gemma-7b

`gemma-7b-it`[8]). Although we filtered for base models in the previous step, not all model cards clearly specify their base model, and models from the same family are often from the same organization. Additionally, model cards from the same organization often follow a consistent structure and writing style, even across different models (e.g., the model card of `Llama-2-7b`[9] and `Meta-Llama-3-8B`[10]). Therefore, to mitigate potential bias from structurally similar model cards, we selected only one model (the top one) from each organization. This left us with 61 model cards.

- **Manually curated model cards:** One author manually removed repositories that are not model repositories (e.g., the `hollowstrawberry/stable-diffusion-guide`[11] repository), do not contain a model card for a single model (e.g., model card of `YoungMasterFromSect/Trauter_LoRAs`[12]), or were incorrectly included due to automation errors (e.g., model card of `CausalLM/14B`[13], which contains non-English text). After this final step, we were left with 48 model cards.

2.2 Information Checklist Preparation

When reorganizing a model card, sentences from the original model card may be split and redistributed across different sections. This fragmentation makes it difficult to verify whether all information from the original model card is preserved in the reorganized version. To facilitate systematic comparison, we introduce an intermediate step that constructs a checklist from the original model card, decomposing each sentence into distinct units of information. For example, we break down the following sentence:

> *Stable Diffusion is a latent text-to-image diffusion model capable of generating photo-realistic images given any text input.*

into these three sentences:

(1) *Stable Diffusion is a latent text-to-image diffusion model.*
(2) *Stable Diffusion is capable of generating photo-realistic images.*
(3) *Stable Diffusion works with any text input.*

After the reorganization of model card contents, the following types of information discrepancies may arise:

- **Missing information:** Information included in the original model card but absent in the reorganized model card.
- **Extra information:** Information included in the reorganized model card but absent from the original model card, typically due to hallucination or misinterpretation. While such statements may seem plausible, they are considered extraneous since they were not explicitly present in the original.

To identify such discrepancies in the reorganized model cards, we used these checklists. By breaking down the original model card at the information level, the checklists enabled a more granular and systematic verification of content preservation in the reorganized versions. However, manually creating the checklists is time-consuming and error-prone. Given the strength of LLMs to understand and process natural language, we used `gpt-4o-mini-2024-07-18` to generate them. The instructions for checklist preparation can be found in our replication package. To verify the checklists,

[8] https://huggingface.co/google/gemma-7b-it
[9] https://huggingface.co/meta-llama/Llama-2-7b
[10] https://huggingface.co/meta-llama/Meta-Llama-3-8B
[11] https://huggingface.co/hollowstrawberry/stable-diffusion-guide
[12] https://huggingface.co/YoungMasterFromSect/Trauter_LoRAs
[13] https://huggingface.co/CausalLM/14B

```
You are an AI assistant tasked with reorganizing model card content to fit a provided template. Your goal is to place existing
    information into the correct sections and subsections without altering or adding any new content.

**Instructions:**

1. Carefully review the provided model card content and the model card template.
2. For each section and subsection in the template, locate the corresponding information within the model card content.
3. Move the information to the appropriate section and subsection in the template. Merge scattered information into the correct
    sections of the template. If a section already exists, you can move relevant details from other parts to complete that section.
4. If a section or subsection in the template does not have corresponding details in the model card content, write "Not available." in
    that section or subsection.
5. If the model card content includes information (even part of a sentence) that does not fit into any section or subsection of the
    template, create a new section titled "Additional Information" at the end of the reorganized model card. Place all extra
    information under this section.
6. Ensure every piece of content (including every details, explanations, reasoning, examples, images, tables, code blocks, citation,
    links, and emojis) from the original model card is present and that no new content is added. Your task is solely to reorganize
    the existing content.

**Model Card Content:** """<the model card>"""

**Model Card Template:**"""<the model card template>"""

**Reorganized Model Card:**
```

Fig. 2. Prompt template used to reorganize a model card's content. The model card template with the section descriptions is available in our replication package.

two authors of this paper manually reviewed the checklists in three rounds, identifying and correcting missing or extra information in the checklists. After each round, they compared their results and resolved disagreements through discussion.

A median checklist had 69 items. We computed the normalized Levenshtein distance between the checklists generated by `GPT-4o mini` and their manually corrected counterparts to evaluate the model's performance in checklist generation. This metric quantifies the extent of edits required to correct the generated checklists. The average normalized Levenshtein distance between the `GPT-4o mini`-generated and manually corrected checklists is 0.03. This indicates that the generated checklists are very close to the manually revised versions, requiring only minimal edits. Such a low distance reflects strong alignment with human expectations, suggesting that the end users of MCTidy can use the checklists directly for their verification too. The usage of the checklists to identify missing (RQ1.1) and extra (RQ1.2) information is detailed in the respective sections.

2.3 Reorganization Using MCTidy

Listing 2 presents the prompt we used to reorganize the content of the 48 model cards with `gemini-2.0-flash-thinking-exp-01-21`, using a temperature setting of 0. We provide a standard model card template along with the original model card in the prompt to guide the reorganization. The template specifies the required sections and subsections, along with descriptions of the expected content for each. The 48 original model cards and the standard model card template, including section and subsection descriptions, are available in our replication package [34].

System Instruction:
A model card is a document designed to provide clear, accessible information about a machine learning model. It helps its users and stakeholders understand how the model works, what data it was trained on, how it should be applied etc.

You will be provided with a model repository. The repository may include various materials such as model development code, academic paper describing the model, and other related resources. Your task is to **write a model card using only the information available in the provided repository**.

All content in the model card must be based on the provided repository data, and **every piece of information used must be accompanied by a citation to its source**. If the provided data do not contain sufficient information for a particular section or subsection, write **"Insufficient information"** for that section or subsection.

Below is the template of a standard model card enclosed in triple quotes. The template contains **11 sections**, some of which include multiple subsections. You need to fill in all sections and subsections comprehensively based on the available information. Each section and subsection contains instructions describing what content to include – replace these instructions with the actual content derived from the repository.
"""<the model card template>"""

User Message:
Write down the model card from the model's attached repository files. Below are the names of the attached repository files:
"""<the model repository file names>"""

Fig. 3. Prompt template used to generate a model card from its content. The model card template with the section descriptions is available in our replication package.

2.4 Generation Using MCGenie

To generate model cards, we first removed the existing model cards from the 48 model repositories to avoid influencing the generation process. The model repository files were then provided to `gemini-2.5-pro` along with detailed generation instructions and the standard model card template prepared in Section 2.3. MCGenie was configured with a temperature setting of 0 to ensure deterministic outputs. The full prompt used for generation is presented in Listing 3.

To prepare the model repository files for input to MCGenie, we cloned the Hugging Face repository of each model, excluding files stored in Git Large File Storage (LFS) and listed the files. Since the `Gemini 2.5 Pro` model does not support files with `.svg` and `.gif` extensions, we removed those from the list. We further excluded files that were not informative for model card generation, such as model parameter files, serialized or sharded model checkpoints, and other large binary artifacts. Finally, we downloaded the associated academic or white papers available for each repository and included them in the list.

Then, we provided the listed model repository files for each model to `Gemini 2.5 Pro` for corresponding model card generation. We successfully generated 26 model cards without exceeding the maximum input token limit of `Gemini 2.5 Pro`. For the remaining 22 repositories that produced errors, we examined common sources of large input files and found that many contained a large `tokenizer.json` file. Since a `tokenizer.json` file is informative for model card generation, we created summarized versions of it and resubmitted the inputs to MCGenie. We summarize each tokenizer JSON file by extracting its model type, vocabulary size, and small samples of vocabulary entries and merge rules. We also retain the normalization, pre-tokenization, post-processing, and decoding configurations, along with all explicitly marked special tokens. This produces a compact representation that preserves essential tokenizer characteristics while reducing file size.

Additionally, two repositories, `deepseek-ai/DeepSeek-V2-Chat` and `facebook/seamless-m4t-v2-large`, included large `model.safetensors.index.json` and `generation_config.json` files, respectively. We similarly summarized key information from these files for inclusion. Thus, we could generate 18 more model cards without the error.

Among the remaining 4 model repositories, we found 3 repositories, `Qwen/Qwen2-VL-7B-Instruct`, `Tencent-Hunyuan/HunyuanDiT`, `openai/whisper-large-v3`, containing large `vocab.json` files. Since the `tokenizer.json` file partially includes the information from `vocab.json`, we removed these vocabulary files from the file list. In the last repository, `ctheodoris/Geneformer`, we identified a large notebook, containing example usage, that significantly increased the total input token count. Although this resulted in some information loss, the notebook was removed to fit within the input constraints.

3 Evaluation of MCTidy

In this section, we present the evaluation of our LLM-based model card reorganization approach, MCTidy.

3.1 RQ1.1: How correct are the reorganizations performed by MCTidy?

Motivation: Since LLMs may omit or misplace information due to their generative nature, it is important to evaluate (1) how well the reorganized model cards retain all key content from the original model card and (2) whether it is assigned to the appropriate sections.

Approach: To evaluate information retention, we used the corrected model information checklists from Section 2.2. One author manually reviewed each reorganized model card alongside its checklist, marking each item in the checklist as present, partially missing, or missing. All missing items and the missing portions of partially missing items were then manually added to the "Additional Information" section of the reorganized model card to complete it with the full set of information.

We quantified information retention by calculating the percentage of present checklist items per model card. We also computed the Levenshtein distance between each reorganized model card and its corresponding manually completed version, normalized by the length of the completed version. While the percentage of present checklist items quantifies the proportion of documentation information that was captured during the reorganization process, the normalized Levenshtein distance captures the extent of information relative to the full documentation that was not captured.

We also assessed whether longer model cards were more prone to omissions. Therefore, we calculated the Pearson correlation between the number of total checklist items and the number of missing items. We interpreted the strength of the correlation using commonly accepted thresholds [30]: 0.00–0.10 (negligible), 0.10–0.39 (weak), 0.40–0.69 (moderate), 0.70–0.89 (strong), and 0.90–1.00 (very strong). Additionally, we computed the p-value to assess the statistical significance of the correlation, with $p < 0.05$ indicating a significant relationship.

To evaluate whether checklist items were placed into the appropriate model card sections, we used an LLM jury [37], an increasingly popular method [15, 18, 31] that simulates multiple independent evaluators and aggregates their judgments (e.g., through majority voting) to reduce bias and improve robustness. We employed three LLMs: `Gemini-2.5-Pro`, `OpenAI o4-mini`, and `DeepSeek R1` as jury models, selected based on cost analysis and their rankings on the Arena leaderboard [6]. We asked each juror model to answer "Is the content in this section relevant to this section?" as "yes" or "no". If judged irrelevant, the model was also asked to identify the irrelevant parts.

We used majority voting across the LLMs to determine whether a section contained misplaced content. Two authors then manually reviewed each flagged section to validate the models' judgments and confirmed whether the identified content was indeed misplaced based on their judgment. We quantified section appropriateness by calculating the

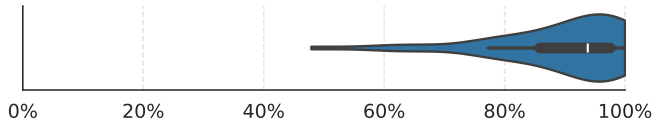

Fig. 4. Distribution of retained checklist items (in percentage) across 48 model cards

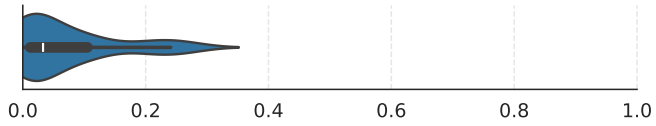

Fig. 5. Distribution of normalized Levenshtein distances between the 48 reorganized model cards and their corresponding manually *completed* versions to measure the proportion of information from the full model card that was not captured

percentage of (sub)sections that contained confirmed misplaced content out of all (sub)sections in the 48 reorganized model cards. This reflects how well MCTidy aligned content with appropriate sections.

Findings: **MCTidy retained a median of 93.8% of the information from the original model card to the reorganized model card.** Figure 4 shows that the best-performing model cards (6) retained all items, while the worst one retained 47.9% of items. The median reorganized model card had only 1.6% of checklist items partially missing and only 4.5% completely missing. We also investigated whether model cards with more checklist items were likely to have more missing items, but the Pearson correlation between the number of checklist items and the number of missing items was not statistically significant ($p = 0.1310$).

Only a small number of edits were needed to manually complete most of the reorganized model cards. As shown in Figure 5, the median normalized Levenshtein distance between the reorganized and manually completed cards was 0.03, indicating very low textual deviation. The best-performing reorganizations that retained all checklist items had a normalized distance of 0.0 (no edits were needed). In contrast, the worst-performing reorganization had a normalized distance of 0.28, reflecting substantial missing content relative to the full model card.

Upon closer inspection of the partially and completely missing checklist items, **we observed several patterns of common omissions**. Some omissions were acceptable due to redundancy or low relevance, while others involved important content that ideally should have been retained. The most frequently missing types of content included:

- **Practical usage guidance**: These are tips or examples to help users interact with the model, such as how to format inputs or understand outputs. For example, the model card for `CAMB-AI/MARS5-TTS` includes tips like adding commas to insert pauses in speech synthesis, and similar formatting guidance appears in `jbetker/tortoise-tts-v2` and `CohereForAI/c4ai-command-r-plus`. These tips are important for the model's end user and should be retained in the reorganized version.
- **Explanatory statements**: These are explanations or descriptions of why certain tasks were done. For instance, the sentence *"Our goal in performing this evaluation was to try to identify ..."* from `HuggingFaceM4/idefics2-8b` was omitted. While not directly related to the model's behavior, such explanations can provide important context and enhance the model's transparency, and are therefore worth retaining.

- **Subjective content**: These are contents like personal opinions, community feedback, or general statements. For instance, informal phrases like *"You may love or hate it"* (e.g., `Crosstyan/BPModel`) or general statements like *"Language models are widely used for tasks other than token prediction"* from `EleutherAI/gpt-j-6b` were excluded. While such content may not be strictly factual or technical, it can sometimes help explain a design choice, provide context, or clarify the intent behind certain decisions. As such, retaining them could improve the readability or interpretability of the model card.
- **Introductory blurbs**: These are short phrases in the `Citation` subsection asking users to cite the model before showing the actual citation. They were often omitted, like in `ibm-granite/granite-timeseries-ttm-v1` and `openlm-research/open-llama-13b`). Since they do not provide additional information beyond the citation itself, their omission does not negatively impact the completeness or utility of the reorganized model card.
- **Navigational or meta-information phrases**: These are references that help users move through the document but do not add specific model-related details. For example, in `openai/whisper-large-v3`, the sentence *"For more details on the different checkpoints available, refer to the section [Model details](#model-details)"* was omitted. Similar references to internal sections were also missing in `NousResearch/Llama-2-7b-chat-hf`, `HuggingFaceM4/idefics2-8b`, and `openchat/openchat_3.5`. Including such phrases in the reorganized version can sometimes create confusion by referring to incorrect or missing sections in the reorganized version, therefore, it is better to omit them.

MCTidy places 98.1% of the content in the correct section. Out of the 1,440 (sub)sections across the 48 reorganized model cards, only 28 (1.9%) contained content that was confirmed to be misplaced in that section. **The misplaced content was distributed across various sections**. As shown in the '# with misplaced information' column in Table 1, no single (sub)section consistently suffered from misplacement. The highest number of confirmed misplaced instances (4) appeared in the `Quantitative Analyses/Unitary results` subsection, followed by four other (sub)sections with three cases each. Given the broad distribution of these instances, we cannot confidently attribute misplacement to any specific (sub)section.

In 10 cases, the LLM jury flagged sections as containing misplaced content, but human reviewers disagreed. The conflict often stemmed from the jury's strict adherence to section descriptions, whereas human evaluators applied a more flexible, context-aware perspective. Jurors also occasionally flagged sections as containing misplaced content due to a lack of expected elaboration. Human reviewers still marked them as relevant, as the content appropriately belonged to the section.

To verify content placement, we also analyzed the amount of information placed in the "Additional Information" section. Since this section can contain any type of content without being considered misplaced into that section, we could not apply the same evaluation process we followed for other sections. Instead, we measured the proportion of content it contains relative to the entire document to assess whether it tends to hold a large share of information, which could impact the accuracy and clarity of content placement within the model card. As shown in Figure 6, the "Additional Information" section is a median of 9.8% of the total model card, with 4 model cards not having an additional information section, indicating that this section generally comprises a small portion of the full model card.

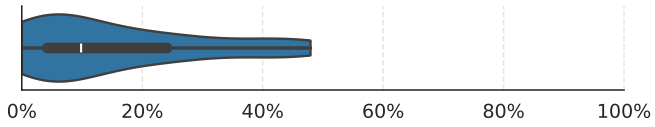

Fig. 6. Distribution of the "Additional Information " section length as a percentage of the total model card length across 48 model cards

MCTidy demonstrates strong overall performance in retaining key content and assigning it to appropriate sections. It achieves a high median checklist retention rate of 93.8% and a low median normalized Levenshtein distance of 0.03, indicating minimal textual omissions. While some missing content, such as usage guidance or subjective explanations, is important, other omissions like introductory blurbs or navigational phrases, have limited effect on overall completeness. Content placement was also highly accurate, with only 1.9% of (sub)sections from all the 48 model cards containing misplaced content, suggesting content misplacement is occasional rather than systematic. Their low frequency and limited impact indicate that minimal human oversight is sufficient to maintain the quality of the reorganized model cards.

3.2 RQ1.2: How much hallucinated content is introduced by MCTidy?

Motivation: As LLMs hallucinate, there can be some extra information that is not present in the original model card. Therefore, we examined the reorganized model cards for added content to evaluate how much new information was introduced during reorganization to understand how much we can trust MCTidy.

Approach: Similar to identifying missing information in reorganized model cards, we used the corrected model information checklists to identify extra information. While reviewing each reorganized model card for completeness, if we found content that was not present in the checklist, we marked it as extra information and removed it from the reorganized version. However, if the generated information was a misinterpretation of the original model card, we corrected it.

Similar to our approach for RQ1.1, we assessed the accuracy of information in the model cards reorganized by MCTidy by computing the Levenshtein distance between each reorganized model card and its manually corrected version, normalized by the length of the reorganized version. We then reported the distribution of these distances across all model cards. This metric captures the overall textual deviation between the reorganized and corrected versions, reflecting the extent of extra information in proportion to the full documentation.

Findings: **Hallucination was minimal across reorganized model cards.** Most model cards require little to no manual editing to remove or correct hallucinated content. As shown in Figure 7, the median normalized Levenshtein distance between a reorganized model card and its corrected version is 0.01, indicating a very low level of textual deviation. The best-performing reorganizations (11) had a normalized distance of 0.0 (no edits were needed). In contrast, the worst-performing reorganization had a normalized distance of 0.19, reflecting substantial incorrect content generated relative to the full model card.

Most section-level corrections were concentrated in a few specific areas. We found hallucinations in a total of 147 (10.2%) (sub)sections out of 1,440 (sub)sections from the 48 model cards. From Table 1 we see that hallucinations occurred most often in the `Primary intended users` subsection from the `Intended Use` section and in the `Ethical`

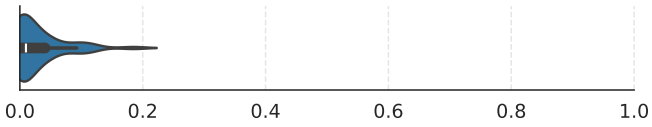

Fig. 7. Distribution of normalized Levenshtein distances between the 48 reorganized model cards and their corresponding manually *corrected* versions to measure the extent of hallucinated information relative to the full model card

`Considerations` section. These sections typically contain descriptive or interpretive content, rather than concrete technical details about the model itself. As a result, they may leave more room for ambiguity or inferred content during reorganization.

We observed recurring patterns in the types of extra information introduced by MCTidy. In the `Primary intended users` subsection, MCTidy introduced extra content by generalizing or expanding the list of user groups in almost each of the 15 cases. For example, it frequently added long lists such as "researchers, developers, businesses, and educators", even when the original model card mentioned only a few of these roles or none at all. This suggests that MCTidy tends to assume a broad audience for models.

In the `Ethical Considerations` section, we observed that MCTidy frequently (in 10 model cards) included the section description verbatim from the provided template. Some model cards also included general ethical or safety statements that were not grounded in the original content. For example, in the `DeepSeek-AI/DeepSeek-V2-Chat` model card, MCTidy added broad claims about safety practices that were not documented in the original model card.

We also found made-up content in the `Motivation` sections of the model cards, specifically, `TrainingData/Motivation` and `Evaluation Data/Motivation`. In these cases, MCTidy introduced motivations that were not present in the original model card. These hallucinated motivations were likely inferred from model characteristics (e.g., size or architecture); however, they lacked direct support from the original model card. Additionally, MCTidy often appended explanatory phrases after inferred content, such as "This implies ..." or "This suggests ..." (e.g., in `openchat/openchat-3.5`), regardless of the (sub)section.

We also observed repeated issues with misinterpretations of referential terms such as "following", "below", or "these". MCTidy often kept these expressions without correctly identifying what they referred to, which led to unclear or misleading references. In some cases, it claimed that certain information could be found in sections that were present in the original model card but did not exist in the reorganized card, or misattributed content to the wrong section. These types of errors created confusion and reduced the usefulness of the model card by pointing readers to content that was not present.

> Only 10.1% of the (sub)sections of the reorganized model cards required minor corrections for extra or misinterpreted content. Hallucinations occurred the most often in the `Primary intended users` and `Ethical Considerations` sections, indicating that while accuracy is generally high, it can be helpful to conduct a manual review of certain subsections after reorganizing a model card.

Table 1. The number of reorganized model cards with misplaced information (RQ1) or hallucinations (RQ2), and the semantic similarity per subsection (RQ3). For RQ1 and RQ2, darker colors mean worse outcomes (more misplaced information or hallucinations), and for RQ3, darker colors mean better outcomes (higher similarity).

(Sub)sections	RQ1.1 # with misplaced information (out of 48)	RQ1.2 # with hallucinations (out of 48)	RQ1.3 Semantic similarity scores	
			Avg.	Med.
Model Details/Person or organization developing model	1 (2.1%)	8 (16.7%)	0.94	0.97
Model Details/Model date	0 (0.0%)	5 (10.4%)	0.92	0.93
Model Details/Model version	0 (0.0%)	4 (8.3%)	0.92	0.92
Model Details/Model type	0 (0.0%)	4 (8.3%)	0.95	0.95
Model Details/Training details	3 (6.2%)	2 (4.2%)	0.95	0.96
Model Details/Paper or other resource for more information	1 (2.1%)	5 (10.4%)	0.93	0.95
Model Details/Citation details	0 (0.0%)	0 (0.0%)	0.99	1.00
Model Details/License	0 (0.0%)	4 (8.3%)	0.98	1.00
Model Details/Contact	0 (0.0%)	1 (2.1%)	0.97	1.00
Intended Use/Primary intended uses	2 (4.2%)	4 (8.3%)	0.96	0.96
Intended Use/Primary intended users	0 (0.0%)	15 (31.2%)	0.92	0.94
Intended Use/Out-of-scope uses	0 (0.0%)	1 (2.1%)	0.96	0.97
How to Use	0 (0.0%)	4 (8.3%)	0.97	0.98
Factors/Relevant factors	0 (0.0%)	4 (8.3%)	0.89	0.90
Factors/Evaluation factors	3 (6.2%)	3 (6.2%)	0.93	0.91
Metrics/Model performance measures	3 (6.2%)	5 (10.4%)	0.92	0.92
Metrics/Decision thresholds	0 (0.0%)	1 (2.1%)	0.99	1.00
Metrics/Variation approaches	0 (0.0%)	3 (6.2%)	0.95	1.00
Evaluation Data/Datasets	1 (2.1%)	8 (16.7%)	0.93	0.94
Evaluation Data/Motivation	3 (6.2%)	7 (14.6%)	0.90	0.89
Evaluation Data/Preprocessing	2 (4.2%)	2 (4.2%)	0.97	1.00
Training Data/Datasets	0 (0.0%)	5 (10.4%)	0.95	0.98
Training Data/Motivation	1 (2.1%)	8 (16.7%)	0.86	0.85
Training Data/Preprocessing	0 (0.0%)	4 (8.3%)	0.96	0.98
Quantitative Analyses/Unitary results	4 (8.3%)	2 (4.2%)	0.93	0.95
Quantitative Analyses/Intersectional results	0 (0.0%)	2 (4.2%)	0.98	1.00
Memory or Hardware Requirements/Loading Requirements	1 (2.1%)	2 (4.2%)	0.93	0.93
Memory or Hardware Requirements/Deploying Requirements	0 (0.0%)	3 (6.2%)	0.93	0.95
Memory or Hardware Requirements/Training or Fine-tuning Requirements	0 (0.0%)	4 (8.3%)	0.96	0.97
Ethical Considerations	1 (2.1%)	15 (31.2%)	0.94	0.95
Caveats and Recommendations/Caveats	0 (0.0%)	7 (14.6%)	0.88	0.88
Caveats and Recommendations/Recommendations	2 (4.2%)	5 (10.4%)	0.89	0.89
Across all (sub)sections	**Total: 28 (1.9%)**	**Total: 147 (10.2%)**	**Avg: 0.94 Med: 0.94**	**Avg: 0.95 Med: 0.95**

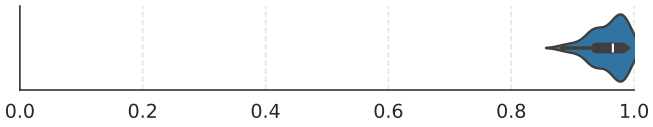

Fig. 8. Distribution of document level semantic similarity scores across 48 model cards

3.3 RQ1.3: How consistent is MCTidy across multiple runs?

Motivation: As LLMs are inherently non-deterministic, the contents in the model cards can be reorganized differently in each run. Therefore, we examined the consistency of content reorganization in model cards to assess whether MCTidy's performance remains stable across runs or exhibits notable variation. This analysis provides insight into the impact of stochasticity on content reorganization.

Approach: We evaluated the consistency of model card reorganization by measuring semantic similarity across multiple runs. We ran MCTidy three times using the same settings and compared the resulting model cards (1) as complete documents and (2) at the level of individual sections. We prioritize semantic similarity over lexical similarity, as our goal is to assess whether contents convey the same meaning rather than use similar wording.

To measure semantic similarity, we used cosine similarity between sentence embeddings generated by the `allenai/specter` [7] model, which is fine-tuned for capturing semantic similarity in scientific text. For each model card, we calculated cosine similarity scores between each pair of reorganized versions: run 1 vs. run 2, run 2 vs. run 3, and run 1 vs. run 3. We then averaged these three scores to obtain a single semantic similarity score per model card (hence 48 scores in total). From this distribution, we computed the overall average and median semantic similarity across all model cards, which reflects the general consistency of the reorganization process in preserving the semantic content across multiple runs. Cosine similarity scores near 1 represent identical or highly similar sentences across runs, and scores near 0 represent no semantic similarity [1].

Findings: **We observed a high degree of consistency in the reorganized model cards across multiple runs**. From Figure 8, we see that the median of the average semantic similarity scores across the three runs is 0.97, indicating that MCTidy reliably preserved the overall meaning of the model cards across runs.

Our section-level analysis revealed that MCTidy produced high semantic consistency across multiple runs. From Table 1, we see that most (87.5%) (sub)sections achieved semantic similarity scores ≥ 0.90. In particular, structured or well-scoped sections such as `Citation details`, `License`, `Contact`, `How to Use`, and `Intersectional results` exhibited near-perfect semantic similarity (≥ 0.98 median, ≥ 0.97 average), indicating that MCTidy consistently preserved the intended meaning of content in these sections. These sections are very straightforward and some are typically short, which likely contributes to their high stability. The `Decision thresholds` and `Evaluation Data/Preprocessing` sections also show near-perfect semantic similarity, but these are mostly empty ("Not available.").

The reorganization process demonstrated high consistency, with a median of the average semantic similarity scores across three runs is 0.97 for the full documents. Most (87.5%) (sub)sections achieved semantic similarity scores of 0.90 or higher, indicating that MCTidy reliably preserves meaning across runs. This high level of consistency suggests that reorganization by MCTidy is not a product of random generation but rather reflects stable and intentional output patterns, strengthening confidence in the approach's reliability for practical model card reorganization task.

3.4 Discussion

MCTidy is a highly viable solution for real-world use in standardizing model card documentation with only limited human oversight. The accuracy of content placement into the correct sections is high, with few hallucinations and misplacements. Also, the reorganizations produced by MCTidy remain highly consistent across multiple runs. These findings collectively indicate MCTidy's stable and reliable behavior, making it a dependable tool for automated model card reorganization.

Across the 48 reorganized model cards, manual effort was required in two areas: (1) completing missing information based on original content, and (2) correcting hallucinated or misinterpreted content. For completion, a median of only 4.5% of checklist items had to be manually added, with a median normalized Levenshtein distance of 0.03 between the reorganized and completed versions, indicating light textual addition. While adding some of the missing information, such as usage guidance, explanatory or subjective content, contributes to user understanding and effective model use, some was not always strictly necessary, such as introductory blurbs and navigation phrases.

In terms of hallucinations, only 147 out of 1,440 (sub)sections (10.2%) needed manual edits. Most of these hallucinations occurred in descriptive or subjective sections, such as `Primary Intended Users` and `Ethical Considerations`, where the model occasionally inferred roles or intentions not explicitly stated in the original model card. These additions were typically vague and speculative, rather than technical, and rarely introduced factual inaccuracies. As a result, they posed minimal risk to the integrity of the documentation and were generally easy to identify and correct.

During our manual analysis of the reorganized model cards, we observed that content placed under the "Additional Information" section was often more appropriately aligned with existing standard sections. In particular, images were frequently relegated to this catch-all section rather than being contextually integrated into the relevant parts of the model card. These findings suggest that MCTidy adopts a conservative strategy when encountering uncertain or partially understood content. While the "Additional Information" section helps avoid incorrect placements, it may reduce the clarity and utility of the reorganized model card. Future improvements could involve systematically further prompting MCTidy to reassess and reassign such content to the most appropriate predefined sections within the model card structure. Recent works [5, 16, 40] support the idea that iterative prompting (asking the LLM to reconsider or refine its initial answer) consistently yields better results than single-shot prompting.

4 Evaluation of MCGenie

In this section, we present the evaluation of our LLM-based model card generation approach, MCGenie.

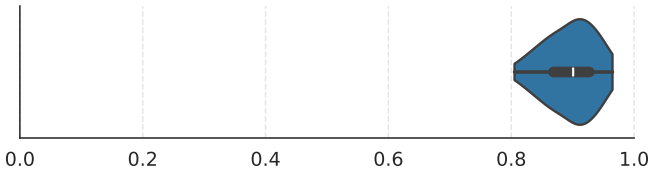

Fig. 9. Distribution of semantic similarity scores of the 48 generated model cards with corresponding original (reorganized) model cards

4.1 RQ2.1: How semantically similar are the generated contents to the original model cards?

Motivation: This will help us understand how close the information in the generated model cards is to the information in the original model cards, to understand how much MCGenie can accurately reproduce or infer key details when generating model cards from available repository data.

Approach: We evaluated the closeness of the generated model cards to the corresponding original model cards by measuring the semantic similarity between the generated and reorganized model cards. Like in Section 3.3, we compared the model cards (1) as complete documents and (2) at the level of individual sections. We used cosine similarity between sentence embeddings of the model cards generated by the `allenai/specter` [7] model to measure semantic similarity. For each model, we calculated the cosine similarity between the generated and reorganized model cards, and then computed the overall average and median similarity scores across all models to reflect their general semantic closeness. Cosine similarity scores close to 1 indicate that the sentences in the model cards are highly similar or nearly identical, whereas scores close to 0 indicate little to no semantic similarity [1]. We further examined the extent to which sections have empty or missing (marked as `Insufficient information` during generation) in the generated model cards, irrespective of having content in the original model card.

We did not compare the original model cards with their reorganized counterparts using semantic similarity because the reorganization step is structure-preserving rather than content-generating. The goal of reorganization is to redistribute existing information across a standardized template without introducing, removing, or modifying content. As such, the original and reorganized model cards are expected to be semantically equivalent by content, differing only in layout and section placement. Consequently, a semantic similarity evaluation between these two versions would be uninformative and potentially misleading, as any measured differences would primarily reflect structural changes rather than content discrepancies. In contrast, comparing the original model cards with the model cards generated from repository data directly evaluates the model's ability to recover and synthesize model card content, which is the primary objective of our evaluation.

Findings: On average, **the generated model cards exhibited a high degree of semantic similarity to the original model cards**. From Figure 9, we can see that when the generated model cards are compared as complete documents with the corresponding original model cards, the mean cosine similarity was 0.9, with a median of 0.9, indicating that the generated model cards generally preserved the key informational content of the originals.

At the section level, similarity varied across sections. Across all (sub)sections, the cosine similarity scores ranged from 0.62 (minimum similarity across all (sub)sections) to 0.99 (maximum similarity score across all (sub)sections), and the average of the average similarity scores for each (sub)section across the 48 model cards was 0.84. Empty sections or sections with limited content, such as `Decision thresholds`, `Intersectional results`, `Citation details`,

and `Variation approaches`, achieved the highest similarity scores (averaging 0.88–0.91), likely because both the original and generated model cards contained minimal or no substantive information in these sections. In contrast, sections requiring interpretation, synthesis, or contextual reasoning, such as `Recommendations`, `Motivation` (under both `Evaluation Data` and `Training Data`), and *Caveats*, showed comparatively lower similarity scores (averaging 0.76–0.80). This indicates that, while these sections are inherently more challenging to replicate fully, the model cards still capture a significant portion of the content from the original model card. Importantly, this highlights both the practical utility of automated generation and the opportunity to further improve coverage in these nuanced sections through targeted guidance, structured templates, or supplementary prompts.

When we analyzed the empty sections in the generated model cards, from Table 2, we observed that, compared to the reorganized model cards, the number of empty sections in the generated model cards increased for 16 (sub)sections and decreased for 13 (sub)sections. In all cases, the changes are less than 30 percentage points, except for the `Evaluation Data/Preprocessing` subsection, where the proportion of empty subsections decreased by 33.3 percentage points.

> The generated model cards showed a high degree of semantic similarity to the original ones, with a median cosine similarity score of 0.9. However, section-level analysis revealed a pattern: sections containing limited or straightforward data tended to have higher similarity, whereas sections requiring interpretation, synthesis, or contextual reasoning exhibited only moderate to high similarity.

4.2 RQ2.2: How correct is the information in the generated model cards?

Motivation: While in RQ2.1, we evaluated semantic similarity between the original and generated model cards to assess overall content preservation, high similarity does not guarantee factual correctness. Large language models can produce text that is semantically coherent and closely aligned with the source while still introducing unsupported, speculative, or fabricated details. Therefore, in this research question, we focus specifically on the factual correctness of the generated model cards. We examine the extent to which information in the generated cards is explicitly supported by the original repository files, helping to assess the factual reliability of MCGenie.

Approach: For each generated model card, we used an LLM jury to evaluate whether the information in each (sub)section of the model card was factually correct based on the corresponding model repository files. We used `Gemini 2.5 Pro` and `Claude Sonnet 4.5` as the two primary jurors because of their extended context window capabilities (approximately 2 million and 1 million tokens, respectively). In cases of disagreement between these two jurors, we used `GPT-5` (supporting approximately 400,000 tokens) as a third juror to resolve the conflict. All three jurors are listed within the top 10 positions on the Text Arena leaderboard[14]. The context window size matters for this evaluation, as the models need to search all provided model resources to verify whether generated information was correct.

Initially, we provided the jurors with all model repository files for evaluation. However, some requests exceeded the token limit, resulting in errors. To address this, we instead supplied the jurors only with the specific repository files cited in the generated model cards and asked them to evaluate each section's factual accuracy based on those files.

When a juror marked a section as incorrect, we also asked it to specify what information was incorrect. However, as in Section 3.1, we did not manually verify the information labeled as incorrect. Confirming whether these instances were truly incorrect would require detailed domain expertise and extensive manual inspection of numerous model

[14] https://lmarena.ai/leaderboard/text

Table 2. (Sub)section-wise percentage of empty sections across 48 model cards. Numbers inside "()" indicate the total number of empty counts for the corresponding (sub)section among the 48 model cards. Darker orange colors indicate worse outcomes (a higher number of empty sections generated compared to the reorganized model cards), while darker blue colors indicate better outcomes (a lower number of empty sections generated compared to the reorganized model cards).

(Sub)sections	Reorganized	Generated
Model Details/Person or organization developing model	2.1% (1)	+ 10.4% (6)
Model Details/Model date	37.5% (18)	− 8.3% (14)
Model Details/Model version	10.4% (5)	− 4.2% (3)
Model Details/Model type	0.0% (0)	0.0% (0)
Model Details/Training details	4.2% (2)	0.0% (2)
Model Details/Paper or other resource for more information	4.2% (2)	+ 27.1% (15)
Model Details/Citation details	29.2% (14)	+ 29.2% (28)
Model Details/License	29.2% (14)	+ 22.9% (25)
Model Details/Contact	56.2% (27)	− 12.5% (21)
Intended Use/Primary intended uses	0.0% (0)	0.0% (0)
Intended Use/Primary intended users	29.2% (14)	− 2.1% (13)
Intended Use/Out-of-scope uses	39.6% (19)	− 6.2% (16)
How to Use	0.0% (0)	+ 18.8% (9)
Factors/Relevant factors	31.2% (15)	+ 4.2% (17)
Factors/Evaluation factors	56.2% (27)	− 16.7% (19)
Metrics/Model performance measures	14.6% (7)	+ 16.7% (15)
Metrics/Decision thresholds	93.8% (45)	− 25.0% (33)
Metrics/Variation approaches	70.8% (34)	− 22.9% (23)
Evaluation Data/Datasets	22.9% (11)	+ 12.5% (17)
Evaluation Data/Motivation	60.4% (29)	− 16.7% (21)
Evaluation Data/Preprocessing	75.0% (36)	− 33.3% (20)
Training Data/Datasets	6.2% (3)	+ 29.2% (17)
Training Data/Motivation	35.4% (17)	+ 8.3% (21)
Training Data/Preprocessing	41.7% (20)	− 27.1% (7)
Quantitative Analyses/Unitary results	29.2% (14)	+ 12.5% (20)
Quantitative Analyses/Intersectional results	91.7% (44)	− 6.2% (41)
Memory or Hardware Requirements/Loading Requirements	62.5% (30)	− 25.0% (18)
Memory or Hardware Requirements/Deploying Requirements	58.3% (28)	+ 6.3% (31)
Memory or Hardware Requirements/Training or Fine-tuning Requirements	35.4% (17)	+ 6.2% (20)
Ethical Considerations	27.1% (13)	+ 4.2% (15)
Caveats and Recommendations/Caveats	6.2% (3)	+ 18.8% (12)
Caveats and Recommendations/Recommendations	6.2% (3)	+ 16.7% (11)

repository files. Whenever two jurors agreed that a (sub)section contained incorrect information, we considered that (sub)section incorrect.

We counted the total number of model cards generated without any incorrect (sub)sections in them to understand the capability of MCGenie to produce model cards without any factually incorrect information. We also manually analyzed the incorrect information in the sections to understand the types of inaccuracies in information generation.

Findings: **More than half of the generated model cards were entirely correct, while the remaining ones contained only a small number of factual inaccuracies.** Twenty-six (54.17%) model cards had all correct sections based on the jury result. From Figure 10, we can see that the average number of incorrect sections in the remaining 22 model cards is 1.41 (4.14% of the total number of sections in a model card), with a median of 1.0 and a range from 1 to 3.

A manual analysis of the incorrect information across the generated model cards identified by the jury revealed several recurring patterns in the types and sources of factual inaccuracies. **Most incorrect information (60%) arises**

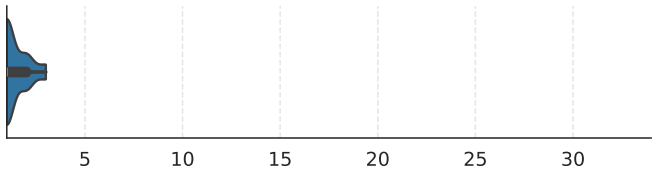

Fig. 10. Distribution of the number of incorrect (sub)sections per model card among the 21 incorrect model cards. Note: One model card has 34 (sub)sections.

from instances where the content in the model card does not accurately reflect the information in the cited sources. These cases typically involve numerical or factual discrepancies between what the model card states and what is found in the associated repository files. Common examples include incorrect dataset sizes (e.g., `BAAI/bge-m3`, `FacebookAI/roberta-large-mnli`), misreported hyperparameters (e.g., `mosaicml/mpt-7b`, `cerebras/Cerebras-GPT-13B`), or conflicting learning rate schedules. Interestingly, some mismatches occur due to information mismatches in the sources themselves. For example, in `openai/whisper-large-v3`, the generated model card states the encoder processes an 80-channel log-Mel spectrogram, citing the associated paper. However, the `config.json` file specifies `num_mel_bins: 128`, indicating it uses a 128-channel spectrogram. This type of mismatch suggests that information consistency in the repository files is a key factor for the factual correctness of the generated model cards.

The second most common inaccuracies result from hallucination (about 23%), where MCGenie makes interpretive or inferential claims not directly supported by the source material. For example, `beomi/llama-2-ko-7b` attributes language specialization to the model name without evidence, and `Tencent-Hunyuan/HunyuanDiT` and `DeepFloyd/IF-I-XL-v1.0` speculate about dataset use or evaluation metrics without textual proof. This suggests that MCGenie tends to "fill in" missing details with plausible but unverifiable statements, a behavior consistent with known LLM tendencies toward hallucination when contextual grounding is weak.

The next pattern, accounting for about 17% of the total, involves cases where the jury indicated that a model card references a document or file that does not exist or cannot be verified. In two instances (8.5%) (`jbetker/tortoise-tts-v2` and `facebook/seamless-m4t-v2-large`), this issue occurred because we had to remove or truncate certain files before providing them to the juror models, as not all file types were supported across all LLMs, resulting in apparent citation mismatches. In two other cases (8.5%), the cited information was drawn from multiple sources that contained conflicting details, making the citation valid for one source but invalid for another. For example, `bigscience/bloom` lists the vocabulary size as 250,880 and cites both `config.json` and the associated paper. While `config.json` confirms the vocabulary size, the paper reports it as 250,680. As a result, citing the paper to support the 250,880 value is incorrect. This finding illustrates that even when the factual content is technically correct, weak or inconsistent citation grounding can compromise transparency and reproducibility, both of which are core objectives of model documentation.

More than half of the generated model cards (54.17%) were fully factually correct. Among the remaining cards, the amount of incorrect information was minimal, as the median model card contained only one (sub)section with an inaccuracy. A manual analysis of these cases showed that, while MCGenie effectively maintains a coherent structure and preserves citations, it sometimes struggles with factual alignment between the generated content and the referenced files. In several cases, this misalignment was caused due to conflicting information in the referenced files.

4.3 RQ2.3: Which input data is the most important for generating model cards?

Motivation: Identifying the most important input data allows users to focus on the relevant repository files when using MCGenie. By providing only the necessary data, users can reduce monetary costs and stay within the model's input token limits, while still ensuring that the generated model card captures the essential information.

Approach: Model repositories contain various files stored in a version control system, and the type of information contained in these files can influence the quality of the information generated in model cards. To examine this impact, we conducted a series of ablation analyses. We first identified the files cited within the generated model cards, as these represent the files actually used by MCGenie during model card generation. We then categorized the cited files by information type and regenerated the model cards using MCGenie under the same settings, but with each information type removed in turn from the repository files. Finally, we compared the regenerated model cards with the originally generated ones and calculated the cosine similarity scores to measure how much they differed.

Our earlier consistency analysis (Section 3.3) showed that reorganized versions of the same model card achieve a very high similarity score of 0.97. Therefore, in this context, we consider similarity scores between 0.97 and 1.0 to indicate that the regenerated model cards preserve the same semantic content as the original generated versions, without information loss.

We also performed a section-level comparison to assess the impact of each information type in more detail. For each (sub)section, we calculated the cosine similarity between the generated and regenerated model cards. In addition, we examined whether removing a particular information type resulted in any (sub)section being empty, which was labeled as "Insufficient Information" during generation. We then analyzed these results across all repositories to identify patterns in how different information types influenced model card quality and to determine which types consistently contributed to more informative sections.

Findings: **Across the analyzed 48 repositories, configuration files (e.g., config.json), tokenizer files (e.g., tokenizer.json), and associated papers were the most commonly cited information sources.** Associated papers exhibited the highest citation rate: all 26 repositories that included a paper consistently cited it in the generated model cards, making it the most frequently referenced information type. Configuration files were present in nearly all repositories (46) and were cited in every case in which they appeared. Tokenizer files were cited less consistently; although 43 repositories contained a tokenizer file, only 34 (79.1%) cited them in the generated model cards. Additional repositories contained tokenizer files stored using LFS due to their large size; these files were excluded from our analysis.

Table 3. (Sub)section-wise semantic similarity score comparison. Darker colors mean better outcomes (higher similarity)

(Sub)sections	w/o Paper		w/o Config		w/o Tokenizer	
	Avg	Med	Avg	Med	Avg	Med
Model Details/Person or organization developing model	0.87	0.84	0.95	0.98	0.94	0.97
Model Details/Model date	0.85	0.86	0.95	0.98	0.93	0.95
Model Details/Model version	0.78	0.77	0.89	0.91	0.91	0.92
Model Details/Model type	0.92	0.93	0.93	0.93	0.96	0.96
Model Details/Training details	0.85	0.85	0.90	0.92	0.92	0.94
Model Details/Paper or other resource for more information	0.75	0.75	0.96	0.98	0.95	0.97
Model Details/Citation details	0.81	0.79	0.96	1.00	0.97	1.00
Model Details/License	0.91	0.95	0.95	0.99	0.96	0.98
Model Details/Contact	0.85	0.83	0.97	0.99	0.98	1.00
Intended Use/Primary intended uses	0.88	0.89	0.93	0.95	0.94	0.95
Intended Use/Primary intended users	0.79	0.79	0.92	0.95	0.92	0.93
Intended Use/Out-of-scope uses	0.84	0.82	0.95	0.97	0.94	0.97
How to Use	0.83	0.80	0.88	0.92	0.88	0.92
Factors/Relevant factors	0.79	0.77	0.94	0.95	0.95	0.96
Factors/Evaluation factors	0.78	0.78	0.95	0.97	0.94	0.96
Metrics/Model performance measures	0.81	0.80	0.97	0.98	0.96	0.97
Metrics/Decision thresholds	0.87	0.86	0.95	0.97	0.97	1.00
Metrics/Variation approaches	0.82	0.83	0.95	0.97	0.94	0.96
Evaluation Data/Datasets	0.79	0.78	0.97	0.98	0.97	0.98
Evaluation Data/Motivation	0.74	0.74	0.95	0.96	0.95	0.96
Evaluation Data/Preprocessing	0.79	0.78	0.93	0.95	0.95	0.98
Training Data/Datasets	0.80	0.79	0.96	0.98	0.95	0.97
Training Data/Motivation	0.73	0.72	0.94	0.96	0.96	0.98
Training Data/Preprocessing	0.85	0.85	0.93	0.96	0.92	0.95
Quantitative Analyses/Unitary results	0.74	0.73	0.95	0.97	0.95	0.97
Quantitative Analyses/Intersectional results	0.89	0.92	0.95	0.98	0.94	1.00
Memory or Hardware Requirements/Loading Requirements	0.89	0.91	0.92	0.95	0.92	0.96
Memory or Hardware Requirements/Deploying Requirements	0.86	0.87	0.93	0.96	0.93	0.98
Memory or Hardware Requirements/Training or Fine-tuning Requirements	0.85	0.85	0.95	0.97	0.94	0.96
Ethical Considerations	0.86	0.86	0.95	0.96	0.96	0.97
Caveats and Recommendations/Caveats	0.77	0.76	0.90	0.93	0.92	0.93
Caveats and Recommendations/Recommendations	0.76	0.75	0.89	0.90	0.90	0.93
Across all (sub)sections	**Avg: 0.82** **Med: 0.82**	**Avg: 0.82** **Med: 0.81**	**Avg: 0.94** **Med: 0.95**	**Avg: 0.96** **Med: 0.96**	**Avg: 0.94** **Med: 0.94**	**Avg: 0.96** **Med: 0.96**

Table 4. (Sub)section-wise comparison of empty sections expressed as percentages. The first value inside "()" demotes the total number of empty counts for the corresponding (sub)section, while the second value denotes the total number of model cards. The empty section counts for the originally generated model cards ('w All Data' column) are same to those reported in the 'Generated' column of Table 2 and are reported here for ease of comparison. Darker colors indicate worse outcomes (more empty sections without the data type in comparison with providing all repository data). For example, the prevalence of darker cells in the 'w/o Paper' column indicate that removing the paper considerably increases the number of empty sections.

(Sub)sections	w All Data	w/o Paper	w/o Config	w/o Tokenizer
Model Details/Person or organization developing model	12.5% (6/48)	+ 29.8% (11/26)	+ 7.1% (9/46)	+ 8.1% (7/34)
Model Details/Model date	29.2% (14/48)	+ 59.3% (23/26)	+ 3.4% (15/46)	+ 3.2% (11/34)
Model Details/Model version	6.2% (3/48)	+ 13.0% (5/26)	+ 6.8% (6/46)	− 3.3% (1/34)
Model Details/Model type	0.0% (0/48)	0.0% (0/26)	0.0% (0/46)	0.0% (0/34)
Model Details/Training details	4.2% (2/48)	− 0.3% (1/26)	+ 21.9% (12/46)	+ 1.7% (2/34)
Model Details/Paper or other resource for more information	31.2% (15/48)	+ 41.8% (19/26)	+ 1.4% (15/46)	+ 1.1% (11/34)
Model Details/Citation details	58.3% (28/48)	+ 41.7% (26/26)	+ 2.5% (28/46)	− 2.5% (19/34)
Model Details/License	52.1% (25/48)	+ 21.0% (19/26)	+ 4.4% (26/46)	+ 9.7% (21/34)
Model Details/Contact	43.8% (21/48)	+ 44.7% (23/26)	− 2.4% (19/46)	+ 6.2% (17/34)
Intended Use/Primary intended uses	0.0% (0/48)	0.0% (0/26)	+ 8.7% (4/46)	0.0% (0/34)
Intended Use/Primary intended users	27.1% (13/48)	+ 49.8% (20/26)	+ 5.5% (15/46)	+ 11.2% (13/34)
Intended Use/Out-of-scope uses	33.3% (16/48)	+ 39.7% (19/26)	+ 10.1% (20/46)	+ 13.7% (16/34)
How to Use	18.8% (9/48)	+ 23.6% (11/26)	+ 3.0% (10/46)	+ 4.8% (8/34)
Factors/Relevant factors	35.4% (17/48)	+ 45.4% (21/26)	+ 5.9% (19/46)	+ 8.7% (15/34)
Factors/Evaluation factors	39.6% (19/48)	+ 48.9% (23/26)	+ 1.7% (19/46)	+ 7.5% (16/34)
Metrics/Model performance measures	31.2% (15/48)	+ 49.5% (21/26)	− 0.8% (14/46)	+ 4.0% (12/34)
Metrics/Decision thresholds	68.8% (33/48)	+ 23.6% (24/26)	+ 3.0% (33/46)	+ 10.7% (27/34)
Metrics/Variation approaches	47.9% (23/48)	+ 48.2% (25/26)	− 0.1% (22/46)	− 0.9% (16/34)
Evaluation Data/Datasets	35.4% (17/48)	+ 53.0% (23/26)	− 0.6% (16/46)	+ 8.7% (15/34)
Evaluation Data/Motivation	43.8% (21/48)	+ 52.4% (25/26)	− 0.3% (20/46)	+ 6.2% (17/34)
Evaluation Data/Preprocessing	41.7% (20/48)	+ 46.8% (23/26)	+ 1.8% (20/46)	+ 5.4% (16/34)
Training Data/Datasets	35.4% (17/48)	+ 30.0% (17/26)	+ 3.7% (18/46)	+ 5.8% (14/34)
Training Data/Motivation	43.8% (21/48)	+ 56.2% (26/26)	− 2.4% (19/46)	+ 6.2% (17/34)
Training Data/Preprocessing	14.6% (7/48)	+ 0.8% (4/26)	+ 2.8% (8/46)	+ 20.7% (12/34)
Quantitative Analyses/Unitary results	41.7% (20/48)	+ 46.8% (23/26)	− 4.7% (17/46)	+ 2.5% (15/34)
Quantitative Analyses/Intersectional results	85.4% (41/48)	+ 14.6% (26/26)	− 2.8% (38/46)	− 3.1% (28/34)
Memory or Hardware Requirements/Loading Requirements	37.5% (18/48)	+ 12.5% (13/26)	+ 16.8% (25/46)	+ 6.6% (15/34)
Memory or Hardware Requirements/Deploying Requirements	64.6% (31/48)	+ 27.7% (24/26)	+ 2.8% (31/46)	+ 6.0% (24/34)
Memory or Hardware Requirements/Training or Fine-tuning Requirements	41.7% (20/48)	+ 39.1% (21/26)	− 0.4% (19/46)	− 0.5% (14/34)
Ethical Considerations	31.2% (15/48)	+ 38.0% (18/26)	+ 3.5% (16/46)	+ 9.9% (14/34)
Caveats and Recommendations/Caveats	25.0% (12/48)	+ 28.8% (14/26)	− 7.6% (8/46)	+ 1.5% (9/34)
Caveats and Recommendations/Recommendations	22.9% (11/48)	+ 54.0% (20/26)	− 1.2% (10/46)	+ 9.4% (11/34)

Excluding associated papers during model card regeneration resulted in a noticeable loss of information compared to the originally generated model cards. Regenerating the model cards without these papers resulted in a median document level cosine similarity of 0.91 when compared to the originally generated model cards. At the (sub)section level, as shown in Table 3, we observe that for a median regenerated model card without its corresponding paper, all (sub)sections, except five, have similarity scores below 0.91. From Table 4, we see that, for the sections `Model Details/Citation details`, `Training Data/Motivation` and `Quantitative Analyses/Intersectional results`, MCGenie was unable to generate any information in the absence of the paper, whereas, when all repository data were available, these sections contained insufficient information in 58.3%, 43.8% and 85.4% of the cases, respectively. This suggests that the paper often serves as the primary and most comprehensive source of contextual details.

Next, we found that **excluding the configuration files during model card regeneration resulted in an information loss, though the impact was smaller than that observed when associated papers were omitted.** Regenerating the model cards without these configuration files resulted in a median document level cosine similarity of 0.95 when compared to the originally generated model cards. At the (sub)section level, as shown in Table 3, we observe that for regenerated model cards without configuration files, the median similarity scores range from 0.90 to 1.00. Except for 6 (sub)sections, all the median scores are ≥0.95. Furthermore, from Table 4, we see that the number of empty (sub)sections does not vary significantly when configuration files are removed. This is likely because the presence of the associated paper reduces reliance on configuration files as a primary information source.

We found that **regenerating the model cards without tokenizer files had a similar impact to regeneration without configuration files.** Excluding tokenizer files resulted in a median document level cosine similarity of 0.96, closely matching the similarity observed when configuration files were omitted. Consistent with this result, the (sub)section-level analysis revealed comparable patterns between model cards regenerated without tokenizer files and those regenerated without configuration files.

> The corresponding papers of the repositories have the greatest impact on the content generation of MCGenie. While configuration and tokenizer files are also frequently cited, their absence has relatively little effect on MCGenie's output, likely because the presence of the associated paper already provides overlapping or more comprehensive information.

5 Threats to Validity

Internal Validity: The first two authors of this paper manually reviewed and corrected the reorganized model cards and their corresponding checklists. Although we aimed to reduce subjectivity by having the authors work independently, the manual process remains inherently subjective and may vary depending on how each individual interprets the information.

An internal threat to validity arises from inconsistencies and ambiguities in the repository files used to generate the model cards. Information about model characteristics is often scattered across multiple sources, which may be outdated or mutually inconsistent. As a result, the generated model cards may contain conflicting or incomplete information, potentially affecting their accuracy.

Construct Validity: We assumed that model cards from top repositories on HF represent a high-quality baseline. Liang et al. [19] showed that the top 0.3% (100 out of 32,111) repositories in HF have good model cards. In our study, to keep our dataset manageable for manual verification, we picked the top 0.1% (1,000 out of 944,178) models. However, the

quality of documentation within these repositories may still vary, and relying on them as our evaluation reference may have introduced bias.

The checklist items used to assess completeness vary in granularity. While most are sentences focused on a single piece of information, some are longer and contain multiple pieces of information. To address this inconsistency, we permitted partial matches during manual evaluation. However, the interpretation of partial completeness remains subjective and may introduce variability in the scoring process.

Further, to reduce human effort and scale the evaluation, we employed a jury of LLMs to assess the correctness of content placement within each section (MCTidy) and the correctness of the content generation (MCGenie). While LLM juries offer a scalable and cost-effective way to simulate multi-rater evaluation, relying on them introduces potential threats to validity. Since LLMs are prone to hallucination and exhibit inherently non-deterministic behavior, this can reduce the reliability and reproducibility of the evaluation. Moreover, LLMs can lack deep comprehension and may rely on superficial patterns or heuristics, leading to assessments that diverge from expert human judgment. In addition, model biases can affect decision quality, especially in technical or context-sensitive scenarios. Finally, because LLMs offer limited transparency in their reasoning, it is difficult to audit their decisions or resolve disagreements. To mitigate these risks, we incorporated human validation, however, the initial use of LLMs in the evaluation process remains a potential threat.

Reorganized content is sometimes placed in the "Additional Information" section by MCTidy when it is unsure where to place that content. While this fallback helps avoid losing information, it can result in too much content being concentrated in that section. This behavior poses a threat to construct validity, as it may distort the intended organization of the content.

Finally, our study applied a uniform evaluation approach across all models, regardless of their type or application domain. Prior work by Richards et al. [29] highlights that model documentation practices can vary based on domain and model type, with some domains requiring more detailed or specialized information. By using a single standard model card template, we may have underrepresented domain-specific documentation patterns, potentially overlooking nuances that are important in certain specialized contexts.

External Validity: Our ground truth for evaluating MCTidy and MCGenie was based on a dataset of 48 model cards, which is not statistically representative of the broader population. This limited and selective sample could introduce bias that restricts the generalizability of our findings. In particular, our evaluation provides a strong indication of MCTidy's capabilities, although their performance on less curated or lower-quality model cards, which are more common in the broader ecosystem, may vary. Also, the capability of MCGenie was evaluated based on the repository files available in these 48 repositories. However, in a broader ecosystem, the types of resources contained in repertoires may vary and could influence MCGenie's performance differently. These variations pose a potential threat to the external validity of our results.

6 Related Work

We group the related work by purpose and present it in the following two subsections.

6.1 Model Documentation Analysis and Improvement

Many studies examine the content of existing model documentation to identify gaps and the scope of improvements in model documentation. Bhat et al. [4] and Liang et al. [19] showed that model cards do not often contain enough

information about different sections of the standard template. Pepe et al. [26] highlighted the need for better documentation of training datasets, biases, and licenses in pre-trained models to improve transparency and mitigate potential biases and legal issues. Oreamuno et al. [24] found that many models and datasets in the Hugging Face store lack comprehensive documentation, either failing to meet user needs or lacking enforcement. The study demonstrated inconsistencies in ethics and transparency-related documentation for ML models and datasets, indicating the need for improved practices to address ethical concerns, biases, and limitations. Additionally, they suggested adding categories for model versioning and attribution in documentation standards. Gao et al. [10] particularly investigated how ethical aspects are currently documented in model cards and provided guidelines for improvement. Through thematic analysis of 256 model cards, they identified six key themes, with the most common being model behavioral risks, intended use cases, and risk mitigation strategies.

Several studies have proposed recommendations for improving model documentation based on interviews, surveys and experience. To enhance ML accountability, Zainyte et al. [38] suggested implementing causality (the effect of model components on the system), decision provenance, and computational tests. They also emphasized the need for detailed factual information in places such as documentation to improve transparency and, in turn, accountability. Nunes et al. [23], through a qualitative study with developers, found that participants were selective about which ethical issues they documented and were generally hesitant to give full autonomy to the models they developed. Piorkowski et al. [27] proposed nine quality dimensions from prior work on software documentation quality analyses and evaluated their usefulness in identifying the quality of AI documentation through a survey.

6.2 Model Documentation Generation

Several recent studies have focused on supporting the creation of model documentation. Fang et al. [9] introduced the Model Card Toolkit (MCT), a set of tools designed to help developers compile and organize the information required for model cards. The toolkit also aids in creating user-friendly interfaces tailored to different audiences. They developed a Python library that can automatically generate model card content based on a predefined JSON schema, using metadata stored in ML Metadata [12]. Similarly, Bhat et al. [4] highlighted the need for better tool support in creating model cards and developed DocML, a tool to help data scientists generate key sections of model documentation more easily in the computational notebook environment.

Some studies presented other ways of representing model-related information. Richards et al. [28] described a user-centered methodology for creating a specific form of AI documentation, named FactSheets [3]. In their 7-step manual design principles, they proposed a set of questions in each step for different stakeholders to identify the information to be included in the FactSheet and progressively build it. Tsay et al. [36] introduced AIMMX, an Artificial Intelligence Model Metadata Extractor, which automatically extracts high-level contextual information from model repositories. Their goal was to address common challenges in documenting AI models, such as the reliance on manual effort and the absence of standardized documentation practices, which often result in inconsistencies and missing information. They also built a searchable catalog using their metadata extractor to support scalable model discovery and management. Crisan et al. [8] introduced and explored the concept of the Interactive Model Card (IMC) as a more accessible and informative way to document deep learning models, with input from ML/AI experts and non-experts. They conducted semi-structured interviews using a think-aloud protocol with 10 ML and AI experts to gather information and feedback on the design of the IMC and implemented a functional prototype of an IMC based on it. They further performed an evaluative study of the functional prototype with 20 non-expert analysts to test the usability and effectiveness of IMC in comparison to traditional standard-model cards.

All these studies operate in a limited setup, such as relying on metadata stored in ML Metadata, using computational notebooks, or focusing on alternative documentation formats. In contrast, our study aims to develop a more widely adoptable approach that aligns with popular and commonly used documentation formats.

7 Conclusion

In this paper, we proposed an automated approach for generating model card documentation that follows a standard template, leveraging resources that are often naturally generated during the model development process, such as model repository files or academic papers. First, we proposed and systematically evaluated MCTidy, an LLM-based reorganization approach for fitting the contents of a model card into a standard template, thereby improving its structure and clarity. By applying MCTidy to 48 model cards from Hugging Face, we demonstrated that our LLM-based approach effectively retains most of the original content (a median of 93.8% of the checklist items), and places it mostly in the correct section (with only 1.9% of (sub)sections of all the 48 model cards having misplaced contents). Additionally, hallucinated content was rare, affecting only 10.1% of (sub)sections from all the 48 model cards, and typically appearing in non-technical areas. We also found that MCTidy's outputs were stable across multiple runs, with a median semantic similarity score of 0.97 for the full documents and at least 0.90 for 87.5% of the (sub)sections. These results collectively highlight the feasibility of using MCTidy to automate model card standardization at scale, with minimal human oversight. To support future research and practical adoption, we released MCTidy and the 48 reorganized and manually curated model cards in a public replication package [34].

Beyond reorganization, we introduced MCGenie, an LLM-based model card generator, to automatically generate new model cards from repository data. Using the reorganized, validated dataset as ground truth, MCGenie achieved high semantic similarity to the reference content at the document level (mean around 0.9) and produced substantial portions of content with factual correctness. Specifically, 54.17% of generated model cards were entirely correct, and most others contained only a small number of minor inaccuracies. The remaining inaccuracies mainly stemmed from information mismatches between cited content and generated content, or speculative reasoning, while the absence or variation of some input files (especially associated papers) strongly influenced generation quality. Section-level similarity was lower for sections that require interpretation or synthesis (roughly 0.76–0.80) than for more data-driven sections (e.g., `Citation details`, `Training data`, and `Intersectional results` reached around 0.88–0.92). Ablation analyses showed that regenerating without the associated paper reduced document-level similarity to about 0.91, whereas removing configuration or tokenizer files reduced similarity to about 0.95–0.96, indicating that papers often serve as the primary grounding source. These results demonstrate the potential of MCGenie to support scalable model card generation, while underscoring the need for high-quality source materials. To foster reproducibility, we released MCGenie resources alongside the reorganized data in the replication package [34].

Together, our findings indicate that LLM-based reorganization and generation can enable scalable, low-effort creation and standardization of model card documentation, with minimal human-in-the-loop requirements.

References

[1] Muhammad Abbas, Alessio Ferrari, Anas Shatnawi, Eduard Enoiu, Mehrdad Saadatmand, and Daniel Sundmark. On the relationship between similar requirements and similar software: A case study in the railway domain. *Requirements Engineering*, 28(1):23–47, 2023.

[2] Shamima Ahmed, Muneer M Alshater, Anis El Ammari, and Helmi Hammami. Artificial intelligence and machine learning in finance: A bibliometric review. *Research in International Business and Finance*, 61:101646, 2022.

[3] Matthew Arnold, Rachel KE Bellamy, Michael Hind, Stephanie Houde, Sameep Mehta, Aleksandra Mojsilović, Ravi Nair, K Natesan Ramamurthy, Alexandra Olteanu, David Piorkowski, et al. FactSheets: Increasing trust in AI services through supplier's declarations of conformity. *IBM Journal of*

Research and Development, 63(4/5):6–1, 2019.

[4] Avinash Bhat, Austin Coursey, Grace Hu, Sixian Li, Nadia Nahar, Shurui Zhou, Christian Kästner, and Jin LC Guo. Aspirations and Practice of ML Model Documentation: Moving the Needle with Nudging and Traceability. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2023.

[5] Kaiyan Chang, Songcheng Xu, Chenglong Wang, Yingfeng Luo, Xiaoqian Liu, Tong Xiao, and Jingbo Zhu. Efficient prompting methods for large language models: A survey. *arXiv preprint arXiv:2404.01077*, 2024.

[6] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024.

[7] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. Specter: Document-level representation learning using citation-informed transformers. *arXiv preprint arXiv:2004.07180*, 2020.

[8] Anamaria Crisan, Margaret Drouhard, Jesse Vig, and Nazneen Rajani. Interactive Model Cards: A Human-Centered Approach to Model Documentation. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pages 427–439, New York, NY, USA, 2022. Association for Computing Machinery.

[9] Huanming Fang, Hui Miao, Karan Shukla, Dan Nanas, Catherina Xu, Christina Greer, Neoklis Polyzotis, Tulsee Doshi, Tiffany Deng, Margaret Mitchell, et al. Introducing the model card toolkit for easier model transparency reporting. *Google AI Blog*, 2020.

[10] Haoyu Gao, Mansooreh Zahedi, Christoph Treude, Sarita Rosenstock, and Marc Cheong. Documenting ethical considerations in open source ai models. In *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 177–188, 2024.

[11] John W Goodell, Satish Kumar, Weng Marc Lim, and Debidutta Pattnaik. Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis. *Journal of Behavioral and Experimental Finance*, 32:100577, 2021.

[12] Google. Ml metadata. https://github.com/google/ml-metadata, 2019. Accessed: 2025-04-06.

[13] Hafsa Habehh and Suril Gohel. Machine learning in healthcare. *Current genomics*, 22(4):291–300, 2021.

[14] Mohd Javaid, Abid Haleem, Ravi Pratap Singh, Rajiv Suman, and Shanay Rab. Significance of machine learning in healthcare: Features, pillars and applications. *International Journal of Intelligent Networks*, 3:58–73, 2022.

[15] Zachary Kenton, Noah Siegel, János Kramár, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah Goodman, et al. On scalable oversight with weak llms judging strong llms. *Advances in Neural Information Processing Systems*, 37:75229–75276, 2024.

[16] Satyapriya Krishna, Chirag Agarwal, and Himabindu Lakkaraju. Understanding the effects of iterative prompting on truthfulness. *arXiv preprint arXiv:2402.06625*, 2024.

[17] Jasmine Latendresse, Samuel Abedu, Ahmad Abdellatif, and Emad Shihab. An exploratory study on machine learning model management. *ACM Trans. Softw. Eng. Methodol.*, 34(1), December 2024.

[18] Hao Li, Cor-Paul Bezemer, and Ahmed E Hassan. Software engineering and foundation models: Insights from industry blogs using a jury of foundation models. *arXiv preprint arXiv:2410.09012*, 2024.

[19] Weixin Liang, Nazneen Rajani, Xinyu Yang, Ezinwanne Ozoani, Eric Wu, Yiqun Chen, Daniel Scott Smith, and James Zou. What's documented in AI? Systematic Analysis of 32K AI Model Cards. *arXiv preprint arXiv:2402.05160*, 2024.

[20] Luca Papariello. xlm-roberta-base-language-detection (revision 9865598), 2024.

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

[22] José Luiz Nunes, Gabriel DJ Barbosa, Clarisse Sieckenius de Souza, and Simone DJ Barbosa. Using Model Cards for ethical reflection on machine learning models: an interview-based study. *Journal on Interactive Systems*, 15(1):1–19, 2024.

[23] José Luiz Nunes, Gabriel DJ Barbosa, Clarisse Sieckenius De Souza, Helio Lopes, and Simone DJ Barbosa. Using model cards for ethical reflection: a qualitative exploration. In *Proceedings of the 21st Brazilian Symposium on Human Factors in Computing Systems*, pages 1–11, 2022.

[24] Ernesto Lang Oreamuno, Rohan Faiyaz Khan, Abdul Ali Bangash, Catherine Stinson, and Bram Adams. The State of Documentation Practices of Third-party Machine Learning Models and Datasets. *IEEE Software*, 2024.

[25] Ezi Ozoani, Marissa Gerchick, and Margaret Mitchell. Appendix — huggingface.co. https://huggingface.co/docs/hub/model-card-appendix#what-do-you-dislike-about-model-cards, 2022. [Accessed 04-04-2025].

[26] Federica Pepe, Vittoria Nardone, Antonio Mastropaolo, Gabriele Bavota, Gerardo Canfora, and Massimiliano Di Penta. How do Hugging Face Models Document Datasets, Bias, and Licenses? An Empirical Study. In *Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension*, pages 370–381, 2024.

[27] David Piorkowski, Daniel González, John Richards, and Stephanie Houde. Towards evaluating and eliciting high-quality documentation for intelligent systems. *arXiv preprint arXiv:2011.08774*, 2020.

[28] John Richards, David Piorkowski, Michael Hind, Stephanie Houde, and Aleksandra Mojsilović. A methodology for creating ai factsheets. *arXiv preprint arXiv:2006.13796*, 2020.

[29] John T Richards, David Piorkowski, Michael Hind, Stephanie Houde, Aleksandra Mojsilovic, and Kush R Varshney. A human-centered methodology for creating ai factsheets. *IEEE Data Eng. Bull.*, 44(4):47–58, 2021.

[30] Patrick Schober, Christa Boer, and Lothar A Schwarte. Correlation coefficients: appropriate use and interpretation. *Anesthesia & analgesia*, 126(5):1763–1768, 2018.

[31] Hyein Seo, Taewook Hwang, Jeesu Jung, Hyeonseok Kang, Hyuk Namgoong, Yohan Lee, and Sangkeun Jung. Large language models as evaluators in education: Verification of feedback consistency and accuracy. *Applied Sciences (2076-3417)*, 15(2), 2025.

[32] Mina Taraghi, Gianolli Dorcelus, Armstrong Foundjem, Florian Tambon, and Foutse Khomh. Deep learning model reuse in the huggingface community: Challenges, benefit and trends. In *2024 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 512–523. IEEE, 2024.

[33] Tajkia Rahman Toma, Balreet Grewal, and Cor-Paul Bezemer. Answering user questions about machine learning models through standardized model cards. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pages 603–603. IEEE Computer Society, 2025.

[34] Tajkia Rahman Toma, Balreet Grewal, and Cor-Paul Bezemer. Replication Package. https://github.com/asgaardlab/mc-reorganization-and-generation, 2026.

[35] Guido Vittorio Travaini, Federico Pacchioni, Silvia Bellumore, Marta Bosia, and Francesco De Micco. Machine learning and criminal justice: A systematic review of advanced methodology for recidivism risk prediction. *International journal of environmental research and public health*, 19(17):10594, 2022.

[36] Jason Tsay, Alan Braz, Martin Hirzel, Avraham Shinnar, and Todd Mummert. AIMMX: Artificial Intelligence Model Metadata Extractor. In *Proceedings of the 17th International Conference on Mining Software Repositories*, pages 81–92, 2020.

[37] Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*, 2024.

[38] Agne Zainyte and Wei Pang. Challenges and future directions for accountable machine learning. In *CEUR Workshop Proceedings*, volume 2894, pages 40–47. CEUR-WS, 2021.

[39] Aleš Završnik. Algorithmic justice: Algorithms and big data in criminal justice settings. *European Journal of criminology*, 18(5):623–642, 2021.

[40] Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. Progressive-hint prompting improves reasoning in large language models. *arXiv preprint arXiv:2304.09797*, 2023.

A Model Card Sections' Description for LLM

Listing 1. Description of the sections of the model card template proposed by Mitchell et al., including additional sections introduced by Toma et al., to inform LLMs about the template structure. Sections introduced by Toma et al. are highlighted in yellow.

Model Details

This section provides fundamental information about the model, helping stakeholders understand its context and key characteristics.

Person or organization developing model: Describe the individual(s) or organization responsible for the development of the model. If available, provide background information, such as their expertise, affiliations, or previous projects they've worked on, include links to their profiles or official websites for credibility. This information helps stakeholders identify the creators and assess credibility.

Model date: Specify the timeline of the model's development. Mention key milestones, such as when the development began, significant updates, and the final release date. This helps users understand the time frame of the methodologies and datasets used.

Model version: Specify the version of the model, and explain how it differs from other versions. For example, describe improvements, bug fixes, additional features, or architectural changes. This aids in tracking updates and assessing progress. If available, explain how the model differs from other similar models.

Model type: Include every detail about the type of the model, including architecture details with available explanations (e.g., Transformer, Convolutional Neural Network) and the specific category it belongs to, such as text generation, image classification, or reinforcement learning. Highlight its core components and how they work together to achieve its functionality. Also, include the model's size and supported context length if available.

Training details: Provide detail with explanation about the training process, including algorithms used (e.g., supervised learning, reinforcement learning), key parameters and hyperparameters (e.g., learning rate, number of layers), fairness constraints or optimization techniques or any other applied methodologies. Provide enough depth for the reader to understand how the model achieved its current state.

Paper or other resource for more information: Include links to related papers, repositories, technical blogs, documentation or other resources that elaborate on the model. If available, briefly summarize the content of these resources and their relevance to understanding the model.

Citation details: Provide citation formats (e.g., BibTeX) for referencing the model in academic or professional work. This allows users to properly acknowledge the creators.

License: Include all available detail related to the license of the model's usage. Outline what users can and cannot do with the model, highlighting any restrictions. If available, include links to the full license text.

Contact: Provide a contact email or other communication channels for users to ask questions, report issues or provide feedback. If available, include additional resources like forums or FAQs.

Intended Use

This section outlines the intended applications of the model.

Primary intended uses: Describe the primary purposes for which the model was created. State whether it was tailored for specific tasks (e.g., sentiment analysis) or built as a general-purpose tool. Be specific about tasks or domains. Explain the capabilities of the model. Include examples of how the model can be applied in real-world scenarios. If available, explain the input-output structure of the model.

Primary intended users: Identify the target audience for the model. Examples include researchers, developers, businesses, or educators. Describe their expected level of expertise and typical use cases.

Out-of-scope uses: List applications the model is not designed for, including potential misuse cases. Highlight situations where the model might be misapplied. Provide examples of related technologies or contexts that might lead to confusion. Suggest alternative models that are more suitable for those contexts if applicable.

How to Use

This section outlines how to use the model.

Include all details on model usage and its example outputs, including input-output structure, settings, code snippets, explanations, and sample input-outputs. If available, add links to documentation and tutorials for further guidance.

Factors

This section addresses variables that may impact the model's performance.

Relevant factors: List key factors that influence the model's performance, such as demographic variations, environmental conditions, or data collection methods. Explain how these factors were identified and why they are important.

Evaluation factors: Indicate which factors are analyzed and reported during model evaluation. If these differ from the relevant factors, explain why they were selected. For instance, you might focus on accuracy metrics over demographic fairness for a specific evaluation.

Metrics

This section describes how the model's performance is evaluated.

Model performance measures: Discuss the metrics used to assess the model's effectiveness (e.g., accuracy, F1 score, precision, recall). Justify the selection of these metrics and why they are more suitable than others.

Decision thresholds: Describe thresholds used in decision-making (e.g., classifying spam emails) and their rationale. Include any empirical evidence supporting these decisions.

Variation approaches: Explain how performance metrics were calculated or estimated, including any uncertainty measures. For example, describe cross-validation, bootstrapping, or other statistical methods used to ensure robust measurements.

Evaluation Data

This section provides details about the datasets used to evaluate the model.

Datasets: Provide information on the datasets used to evaluate the model. Include every details available about the data like size, diversity, source, public availability or proprietary.

Motivation: Explain why these datasets were chosen for evaluation. Discuss their relevance to the model's intended use and how well they represent real-world scenarios.

Preprocessing: Describe the preprocessing steps applied to the evaluation data in detail with explanation. Examples include normalization, encoding, or filtering. If available, explain how these steps align with the model's design and intended use.

Training Data

This section provides details about the datasets used to train the model.

Datasets: Provide information on the datasets used to train the model. Include every details available about the data like size, structure, features, diversity. If the data is publicly available, include links.

Motivation: Justify the choice of datasets for training, highlighting their suitability for the model's purpose and intended applications.

Preprocessing: Describe the preprocessing steps applied to the training data in detail with explanation. Include examples like text tokenization, image resizing, or outlier removal. Explain how these steps improved training efficiency or accuracy.

Quantitative Analyses

This section presents disaggregated evaluation results.

Unitary results: Present performance results for each individual factors identified in the Factors section. For example, show accuracy rates for different demographic groups or under varying environmental conditions.

Intersectional results: Present performance results across combinations of factors. For instance, analyze accuracy for a specific demographic group within a particular geographic location.

Memory or Hardware Requirements

This section outlines the memory or hardware requirements for loading, deploying, and training the model.

Loading Requirements: If available, specify the memory and hardware requirements (e.g., RAM/VRAM size, disk space, CPU/GPU/TPU) to load the model.

Deploying Requirements: If available, specify the memory or hardware requirements to run and serve the model.

Training or Fine-tuning Requirements: If available, specify the memory or hardware requirements to train or finetune the model.

Ethical Considerations

This section discusses the ethical considerations in model development, including challenges, risks, and solutions.

Specify if sensitive data (e.g., personal information, protected attributes) was used. Identify potential risks associated with the model's application, their likelihood, and severity, especially in critical areas like healthcare or public safety. Describe risk mitigation strategies used during development and risks in model usage, including potential harm and affected groups. If risks are unknown, note that they were considered. Highlight the known model use cases that are especially fraught. Highlight efforts to address these challenges and acknowledge areas requiring further exploration.

Caveats and Recommendations

This section lists unresolved issues and provides guidance for users.

Caveats: List any limitations or areas of concern not addressed earlier. For example: gaps in evaluation datasets (e.g., missing demographic groups), suggestions for future testing or research, ideal characteristics of datasets for further evaluation etc.

Recommendations: Suggest best practices for using the model and areas for further testing. Provide actionable recommendations for users to maximize the model's benefits while mitigating risks.