Enhancing Open-Source Model Adoption Analysis of Hugging Face READMEs and LLM-Driven Documentation Tools

Yinan Wu
North Carolina State University
Raleigh, USA
ywu92@ncsu.edu

Xiangwei Wang
North Carolina State University
Raleigh, USA
xwang258@ncsu.edu

Uchswas Paul
North Carolina State University
Raleigh, USA
upaul@ncsu.edu

Xiang Lan
North Carolina State University
Raleigh, USA
xlan4@ncsu.edu

Abstract

The rapid growth of Artificial Intelligence (AI) and Machine Learning (ML) models has led to widespread sharing of models on platforms like Hugging Face. However, the quality of accompanying README files is often inconsistent, making it difficult for users to understand and adopt these models effectively. This study aims to address these issues by exploring two key questions: (1) Can we create a framework of best practices for README files based on high-quality examples? (2) Can we build a tool using a Large Language Model (LLM) to evaluate and improve README quality?

We analyzed over 200 README files from Hugging Face, categorized their key components into 11 elements, and trained an LLM to identify and assess these elements. We identified common components like model descriptions and usage instructions, as well as less frequently included but important sections such as model limitations and licensing. Additionally, we developed a tool to visualize README quality, providing developers with practical insights for improvement.

This research highlights the importance of clear and comprehensive documentation in supporting the adoption of AI/ML models. Our findings offer a foundation for improving open-source documentation and suggest pathways for creating tools that make model documentation more accessible and useful.

Keywords

Hugging Face, Documentation, Classification, Visualization

1 Introduction

The rapid rise of Artificial Intelligence (AI) and Machine Learning (ML) models has transformed numerous fields, including natural language processing, computer vision, healthcare, and scientific discovery. These models are increasingly integral to both academic research and industrial applications, enabling innovative solutions to complex problems. Open-source platforms like Hugging Face have emerged as critical ecosystems for sharing, developing, and deploying these models, fostering collaboration and innovation across diverse communities. However, the adoption and effective use of these models often depend not only on their capabilities but also on the clarity and comprehensiveness of their accompanying documentation.

README documentation serves as the primary entry point for understanding and utilizing open-source models. They provide essential information, including model descriptions, usage guidelines, and licensing terms. Despite their importance, many README files are inconsistent in quality, lacking critical details or clear organization. Such issues create barriers for users, particularly those new to the field, limiting the accessibility and usability of these models. This disparity in documentation quality underscores the need for systematic approaches to evaluate, improve, and standardize README content.

Our research was guided by two research questions (RQs):

- **RQ1:** Can we identify a framework of best practices from README files in Hugging Face?
- **RQ2:** Can we build a LLM-powered tool to analyze and visualize existing README quality, providing statistics on content inclusion and completeness?

To address these questions, we conducted a comprehensive study of README files from Hugging Face, leveraging both human annotation and machine learning techniques. Our study resulted in the following contributions:

- A deep-depth analysis of README files from over 200 popular Hugging Face models, highlighting the common elements and gaps in documentation quality.
- A taxonomy of key components that define high-quality README files, derived from both manual annotation and quantitative analysis.
- A fine-tuned large language model (LLM) designed to classify and extract key components from README files, with a focus on improving usability and accessibility.
- A visualization tool that provides actionable insights into README quality, offering a pathway for developers to enhance their documentation.

By addressing the challenges of inconsistent and incomplete documentation, this research aims to enhance the usability and adoption of open-source AI/ML models. Furthermore, our findings provide a foundation for future work in automated documentation tools and best practices for the AI community.

2 Research Method

In this section, we present the methodology employed in our study. Our approach consists of a two-stage pipeline: data collection (Section 2.1) and thematic analysis (Section 2.2).

In the data collection stage, we focused on obtaining high-quality README files from the Hugging Face platform by filtering and refining a comprehensive dataset of models. This step ensured that our analysis was based on representative and relevant data.

In the data analysis stage, we segmented and annotated the README files using a structured framework to identify key components and themes. This process involved both qualitative and quantitative methods to derive meaningful insights and validate the categories through iterative refinement.

2.1 Data Collection

We consider README files from Hugging Face as the primary dataset for our study. The data collection process consisted of the following steps:

Metadata Crawling: We began by crawling metadata for 1,049,690 models available on the Hugging Face platform as of October 13, 2024. This metadata provided information such as model names, descriptions, and popularity metrics, which were essential for filtering and selection.

Data Filtering: To focus on high-quality and widely used models, we applied a filtering criterion based on the number of likes. Specifically, we selected the top 5% of models with more than two likes, ensuring that less relevant "toy projects" were excluded. After this step, 2,236 popular models were retained.

API Access: Using the Hugging Face API, we retrieved the corresponding README files for the filtered models. The API enabled efficient access to the documentation content, which served as the foundation for our analysis.

Final Dataset Preparation: After removing inaccessible files, we successfully retrieved 2,057 README files. This final dataset represented a diverse and substantial collection of documentation, suitable for large-scale analysis and model training.

This data collection process ensured that our dataset was both high-quality and comprehensive, providing a solid foundation for the subsequent analysis phase.

2.2 Thematic Analysis

Once we have collected the README files from Hugging Face, we proceed to process this raw data by segmenting it into meaningful units for further analysis. Each README file on Hugging Face is formatted using Markdown, a lightweight markup language that structures text through various annotations and hierarchical headers. Leveraging this standardized structure, we utilize the inherent formatting of the Markdown source code to systematically split the README files into distinct segments according to their header annotations, such as levels defined by #, ##, ###, and so on.

This segmentation allows us to organize the data into hierarchical levels, ensuring that each section of the README is captured and classified accurately. For each identified segment, we preserve its hierarchical context by maintaining a structured pairing of the header and its corresponding content, i.e., header, content. This approach not only facilitates efficient data organization but also enables deeper insights during subsequent analysis, as the headers provide context and semantic cues for the associated content. By following this structured methodology, we ensure the integrity of the segmented data and enhance its usability for downstream tasks, such as natural language processing or documentation analysis

We conducted *thematic analysis* [1], a method particularly suited for identifying and analyzing patterns and themes within qualitative data, to interpret the posts in our dataset.

For post itself. The first step of our analysis is to build familiarity with the dataset and create an initial codebook. We follow a standard open coding method named *Saldana open coding* [4],which is an essential first step in grounded theory or other qualitative analysis methods, where researchers break down qualitative data into discrete parts and label them with codes to identify significant patterns, themes, and categories. The key principles of this open coding method includes:

- *Iterative and Reflective Process*: Coding is not a one-time activity but iterative. Researchers often revisit the data multiple times, refining codes as insights evolve;
- *Descriptive and Interpretive*: Codes can be descriptive (summarizing the explicit content of the data) or interpretive (inferring underlying meanings or implications);
- *Initial Codes are Provisional*: Codes in open coding are not fixed; they serve as starting points. Further analysis may merge, refine, or eliminate these initial codes;
- *Focus on Segments*: Researchers segment the data into smaller units—such as sentences, phrases, or incidents—and apply codes to capture their essence.

In practice, we try to follow these principles to label each segment we get from hugging face. First, each of us annotate 15 samples based on the previous work [3]. Those labels include:

[Model Description]: The Model Description section provides a detailed overview of the AI model, serving as a foundation for understanding its core attributes and functionality. It outlines the architecture, such as GPT, BERT, or ResNet, and highlights the model's primary purpose, whether for natural language processing, computer vision, or other tasks. This section may also delve into the pretraining process, including datasets used and tasks the model was initially optimized for. Additionally, it covers technical specifications like the number of parameters, layers, or unique design elements that set the model apart. By offering these insights, this section ensures users have a clear understanding of what the model is, its origins, and the innovations it incorporates.

[Usage]: The Usage section focuses on practical applications and guidance for deploying the model effectively. It describes the intended use cases, such as text summarization, translation, or object detection, and provides step-by-step instructions for loading and running the model. This includes dependencies, hardware or software requirements, and any necessary setup procedures. To enhance clarity, it may also provide examples of input formats and expected outputs, ensuring users understand how to interact with the model. Additionally, this section highlights any constraints, like token length or resolution limits, helping users manage expectations and avoid misuse.

[Citation/Reference]:The Citation/Reference section ensures proper attribution of the model by providing detailed information on how to cite it in academic or professional work. It typically includes references to research papers, documentation, or GitHub repositories that describe the model in detail. A standardized citation format may be offered for convenience, along with links to resources such as Digital Object Identifiers (DOIs) if available. This section emphasizes the importance of acknowledging the original creators of the model, fostering transparency and academic integrity in its use.

[License]: The License section clearly states the terms and conditions under which the model can be used, distributed, or modified. This includes specifying the type of license, such as MIT, Apache 2.0, GPL, or proprietary, and outlining any restrictions, such as non-commercial use or requirements for attribution. A link to the full license text is often included for users seeking detailed legal information. By articulating these terms, this section helps users understand their rights and responsibilities, ensuring compliance with legal and ethical standards.

[Evaluation/Performance]: The Evaluation/Performance section provides an in-depth analysis of the model' s capabilities and limitations based on benchmarking results. It describes the datasets used for evaluation and the metrics applied, such as accuracy, F1 score, BLEU, or perplexity. Comparative results against similar models or baselines may also be presented, offering a broader context for understanding the model's strengths and weaknesses. Additionally, this section identifies potential limitations, such as biases or scenarios where the model may underperform, and discusses performance trade-offs, such as speed versus accuracy. By offering these insights, it equips users with a realistic perspective on how the model performs in various scenarios.

During the first round of sampling, if we find segments which are not suitable for any existing categories, We should have a discussion to decide whether to create a new category, merge it into existing categories or drop it. Then the labels set increases to include:

[Model Limitations]: The Model Limitations section outlines the known weaknesses and constraints of the AI model, ensuring users have realistic expectations of its performance. This includes detailing areas where the model may underperform, such as specific types of data, languages, or tasks. It also addresses ethical concerns, like biases that might arise from training data, sensitivity to adversarial inputs, or scenarios where the model' s predictions may lack reliability. By clearly presenting these limitations, this section helps users understand the contexts in which the model should or should not be applied and highlights areas for potential improvement.

[Model Deployment]: The Model Deployment section provides guidance on integrating the model into real-world applications, covering both technical and operational aspects. It explains deployment strategies, such as server-based, cloud, or edge computing setups, and offers recommendations for optimizing performance and scalability. This section may also address safety mechanisms, such as monitoring tools for mitigating errors or harmful outputs, and compliance with legal or ethical standards. By providing these details, it ensures users can deploy the model effectively while maintaining robustness and ethical considerations.

[Table of Contents]: The Table of Contents serves as a navigational guide to the model card, listing all the major sections in a clear and organized manner. This section ensures users can quickly locate specific information, such as technical specifications, usage instructions, or licensing details.

Table 1: Number of readmes containing a specific section

Label	Count
Model Description	183
Usage(How to Use)	172
Evaluation/Performance/Results	81
Citation/References	75
training info	71
Model Limitations	56
License	54
contributions and acknowledgement	17
Model Deployment	12
Table of Contents	9
Contact Information	8

[Contributions and Acknowledgments]: The Contributions and Acknowledgments section recognizes the individuals, teams, and organizations that played a significant role in developing the model. This may include researchers, engineers, and contributors to datasets or open-source tools. It also acknowledges funding sources, institutions, or collaborations that supported the project.

[Training Information]: The Training Information section provides detailed insights into how the model was trained, offering transparency about its development process. This includes the datasets used, training duration, hardware specifications, and any specific optimization techniques employed. It may also discuss pretraining and fine-tuning processes, along with data augmentation or preprocessing steps. By presenting this information, this section helps users understand the factors influencing the model's performance and ethical considerations, such as the source and diversity of training data.

[Contact Information]: The Contact Information section provides details on how users can reach out for support, feedback, or inquiries about the model. This may include email addresses, links to discussion forums, or official channels for reporting issues or sharing suggestions. It ensures open communication between the model creators and the user community, fostering collaboration and addressing concerns effectively.

Finally, we use these lables to annonate on 200 samples. The we do an extra round to check each sample is suitable for those lables and make sure there aren't incorrect or toxic samples.

3 RQ1: Can we identify a framework of best practices from README files in Hugging Face?

By employing our method described in Section 2, we developed a taxonomy of presentation. We are curious which sections are prominent in our 200 samples. So we collect the frequency of each label as shown in Table 1 from the most frequent to the least. It is reasonable that most readmes should include at least model description and usage. But for other sections, they are different for different authors. Some authors have good writing styles then they will contain these sections in their readmes, while others may leave those sections out. But we still recommend that it is a good habit to contain necessary sections in the readmes.

Table 2: Taxonomy of section name in Readme files

Section Name	Other Name
Model Description	Model description, Model Details, Model Summary, Model Architecture Updates, Model Description, Model details
Usage(How to Use)	How to use, Usage, Model Usage, How to use the model, Quick Start, Quickstart, How to Get Started with the Model, Tasks
Evaluation/Performance/Results	Evaluation results, Evaluation, Evaluation dataset, Benchmarks, Performance,
Citation/References	Citation, BibTeX entry and citation info, BibTex and citation info, References, Papers and References
training info	Training data, Training Procedure, Training and Evaluation Data, Recipe Details, Training dataset, Training Stats
Model Limitations	Intended Uses & Limitations, Limitations, Responsible AI Considerations, Safety Evaluation and Red-Teaming
License	—
contributions and acknowledgement	—

For the convenience of our model fine-tuning and practice, we decide to include the top **eight** frequent sections from "model description" to "contributions and acknowledgment" as our final taxonomy. As shown in Table 2, We present a comprehensive list of recommended sections to include in a Hugging Face model card to ensure consistency and clarity across documentation. Recognizing that different individuals may choose varying terms to convey the same or closely related concepts, we have also curated a set of alternative options for section headers. [1]Many of these options have been carefully collected, aggregated, or derived from common patterns observed in existing model card samples, ensuring they are both practical and reflective of community standards. By offering multiple choices for headers, we aim to accommodate diverse preferences and writing styles while promoting a cohesive structure for model cards across projects.

4 RQ2: Can we build an LLM-powered tool to analyze and visualize existing README quality, providing statistics on content inclusion and completeness?

4.1 Fine-Tuning Process and LLM-powered tool

We used the 200 readme files for training data that were labeled during the empirical study. ChatGPT-4o was chosen as the base model due to its strong natural language understanding capabilities. During the training session, the system prompt defined the task, specifying eight key components such as "Introduction," "Usage," and "Model Limitations," among others. The user prompt contains the full readme file. The user prompt contained the content of the full readme text. The fine-tuning process was designed to enhance the model's ability to analyze and extract meaningful content from README files of Hugging Face models. The approach not only classified sections as present or absent (binary classification) but also extended the model's understanding to extract the contextual content of each section. For training, a structured JSON format was used to represent the assistant's response, indicating the presence or absence of each component and providing its corresponding content if present. By leveraging this structured data, the model was fine-tuned to identify the presence of components and retrieve their content accurately, enabling a more nuanced understanding of README documentation. Below, the system prompt is shown, and one example of an assistant prompt in train data is shown in Listing 1.

System Prompt

You are an intelligent system that analyzes README files from Hugging Face models. Your task is to extract and classify eight key components from the README content provided below. Identify whether each component is present or absent and provide the content of each present component. Components include: {{List of 8 key section}}

```
{
    "Model Description": {"present": 1, "content": "This model..."},
    "Usage": {"present": 0, "content": ""},
    "Citation": {"present": 0, "content": ""},
    "Model Limitations": {"present": 1, "content": "While the model is...."},
    "Training Info": {"present": 0, "content": ""},
    "License": {"present": 1, "content": "Use pip install..."},
    "Results": {"present": 1, "content": "The model achieves..."},
    "Contribution": {"present": 0, "content": ""}
}
```

Listing 1: Example of assistant prompt during training

Building upon our training data, we developed a web-based tool that accepts a README file as input and outputs a JSON structure consistent with our training data format. This tool enables us to effectively visualize both the present and missing content within the README file and display the content of the included sections.

5 Evaluiation Metrics

We randomly sampled 30 readme files for evaluation that were not used in training and used our tool to get results. We then constructed a confusion matrix to calculate the precision, Recall, and Accuracy of the model. Precision is calculated as $\frac{\text{TP}}{\text{TP+FP}}$, Recall

[1]Please note that in practice users are free to use other names as long as they have the same functionality for these sections.

is calculated as $\frac{TP}{TP+FN}$ and Accuracy is calculated as $\frac{TP+TN}{TP++TN+FN+FP}$. Here, TP is when both the LLM and human assessments confirm the presence of a section. FN arises when a section is present but missed by the LLM. FP refers to a scenario where the LLM detects a section that is not present in the readme. TN is when both the LLM and human assessments agree that the section doesn't exist. The results in Table 3 highlight the model's strengths in identifying content within README files, as well as areas for improvement. The high precision score (0.99) demonstrates that the model is highly accurate in confirming the presence of specific content, with only one instance of misidentification (false positive), suggesting its adeptness at recognizing structured information and ensuring a low rate of incorrect predictions. However, the recall score (0.91) and the presence of 10 false negatives reveal limitations in identifying content embedded within descriptive text or presented in non-standard formats. Specifically, five false negatives were related to the citation section, where citation details were integrated into the broader text rather than being explicitly formatted, indicating the model's reliance on structured patterns such as BibTeX entries or direct links. Three false negatives in the license section occurred due to license information being embedded within descriptions or placed in unconventional locations, reflecting the model's preference for clearly labeled sections. The other two false negatives in the training data section were observed when the details were scattered across multiple sections or blended with other content, highlighting the need for better context awareness and the ability to infer section boundaries without explicit markers. The single false positive, attributed to a generic link being mistakenly identified as part of the citation section, indicates that while rare, the model occasionally overgeneralizes patterns, necessitating stricter criteria to differentiate between unrelated links and actual citations.

6 Discussion

6.1 Limited Data Size

While our fine-tuning efforts yielded promising accuracies, we need to admit the potential biases introduced by the limitations in our dataset size. The relatively small volume of labeled data inherently restricts the diversity and representativeness of the samples, which may lead to skewed or less generalizable results. Specifically, due to constraints in time and budget, we were only able to manually label 200 README samples, selecting them from those with the highest number of likes. While this approach ensured a focus on popular and likely well-curated examples, it may inadvertently exclude less popular but equally significant variations, introducing a preference bias. This limitation highlights the trade-off between practicality and comprehensiveness, and underscores the need for further refinement and expansion of the dataset in future iterations to enhance robustness and fairness in the model' s performance.

6.2 Fine-grained Sections

The eight sections we selected for fine-tuning the model are typically positioned at the highest structural level in a README file, representing the primary categories within a model card. We carefully designed these sections to ensure that their content does not overlap, thereby maintaining a clear and logical distinction between different aspects of the model' s documentation. These sections are intended to serve as the top-level organization of a model card, providing a concise overview of the most critical information.

However, to make the README more comprehensive, readable, and user-friendly, a more fine-grained approach could be beneficial. By breaking down these top-level sections into specific and well-defined subsections, we can provide greater clarity and detail, catering to users who may seek targeted information. For instance, within the broader section labeled "Training Info," splitting it into more focused subsections such as "Training Data," "Training Procedure," "Optimization Techniques," and "Hyperparameters" would allow for a more nuanced presentation of the training process. This level of granularity not only enhances readability but also improves the practical utility of the documentation.

To achieve this level of refinement, it is essential to develop a more precise and detailed list of subsections for each primary section. Such an approach requires careful consideration of the unique aspects and complexities of the model, as well as a balance between comprehensiveness and brevity to avoid overwhelming the reader.

6.3 Automatically Generation Tool

In this project, we fine-tuned the model to classify different sections given a readme file and to tell which sections are missed in the existing file. Based on this work, we believe we can create an automatically generation tool for hugging face model card. We have already designed what a good framework should be like. The next step is to verify and improve the ability of GPT to generate the corresponding contents given the necessary keywords or information. We hope that finally our tool can help users to automatically create model cards on hugging face.

7 Literature Review

Documentation is a critical component of software development and maintenance, providing the necessary resources for developers and users to effectively understand, use, and contribute to software projects. Among the various forms of documentation, README files have special significance because they often serve as the first point of interaction between a developer and a software project. These files provide a snapshot of the project's purpose, installation instructions, usage guides, and often include additional resources or links for further exploration. Their accessibility and central role make README files an important element for new contributors, clarifying project goals, and facilitating collaboration within the developer community.

In the context of open source software and AI model libraries, README files and related documents, such as model cards and dataset cards, play a vital role in fostering user trust, facilitating collaboration, and achieving efficient utilization of software systems. These documents serve as the first point of user interaction, providing critical information about the purpose, functionality, and requirements of the software. A well-crafted README file can act as a bridge between developers and end users, reducing the learning curve and lowering the barriers to entry. It provides clear instructions, usage examples, and insight into software design and functionality to make it easier for users to get started and for contributors to engage more meaningfully with the project.

Table 3: Confusion Matrix

	LLM says Exists	LLM says Not Exist	Total
Exists In README	101 (TP)	10 (FN)	111
Not Exist in README	1 (FP)	128 (TN)	129
Total	102	138	240

In addition, README files and similar documentation are essential for expanding the open source community. They create a shared understanding among different stakeholders, from individual developers to large organizations. In the context of artificial intelligence, model cards and data set cards go beyond basic usage instructions. They address ethical issues, limitations, and performance metrics, ensuring that users understand the context and potential impact of the software. These documents also promote transparency by gaining insight into the decision-making process behind a model or dataset, which is critical in areas such as bias detection and ethical AI use.

Conversely, poorly written or incomplete documentation can pose significant challenges. Users may misunderstand the purpose of the software or misuse its features, resulting in errors or inefficiencies. In a collaborative environment, lack of clarity in documentation can hinder contributions and slow down development. For AI models, inadequate documentation of factors such as training data, intended use, or known limitations can lead to misuse, distrust, and even ethical violations. This highlights the need for structured, thorough, accessible README files and associated documentation to ensure the success and sustainability of software projects.

This literature review examines previous research on README files and their counterparts, focusing on their characteristics, importance, and role in the broader software development ecosystem. By analyzing current practices and identifying gaps in document quality, this review reveals how README files can contribute to project success, both in terms of technology and as a tool for promoting transparency and inclusion in open source and collaborative software environments. In addition, it explores how README documentation practices align with the growing ethical considerations and need for sustainability in software development, particularly in areas such as artificial intelligence and machine learning.

The README file serves as the first point of interaction for users exploring a software project. They provide key insights into the purpose, use, and installation of the software, thus acting as a bridge between developers and users. Comprehensive README files increase the adoption and usability of projects, making them a key component of open source software projects. Researchers have highlighted how README files enhance the onboarding process for contributors and increase the likelihood of long-term project success.

Liang et al. [3] conducted an in-depth analysis of 32,111 model documents hosted on hug Face, providing key insights into the state of AI model documents. Their research highlights the urgent need for comprehensive and standardized documentation, especially for AI models. While model cards have become an industry standard, they observed that the quality of these documents varies greatly. Many lack basic details on key aspects such as environmental impacts, model limitations, and thorough assessments. These omissions reduce the usability and transparency of AI systems, making it difficult for users to assess their suitability and ethical implications. The researchers believe that consistent and detailed documentation practices can play a vital role in increasing user trust and encouraging widespread adoption of AI technologies. They emphasize that proper documentation is not only a best practice, but also a necessity for responsible AI use.

Yang et al. [5] extended this discussion to dataset documentation practice by analyzing 7,433 dataset cards on hug Face. Their findings highlight the importance of high-quality data set documentation in ensuring ethical and effective data use. Popular datasets are often better documented, providing a comprehensive description of the structure, features, and intended use cases. However, many dataset cards fail to address key issues such as licensing, data sourcing, and ethical constraints. This inconsistency poses challenges for developers and researchers who rely on these datasets to build and evaluate models. Yang et al. call for a more structured and inclusive documentation framework to promote transparency and ethical use, especially for less popular datasets that often lack sufficient detail.

Gao et al. [2] focuses on the often overlooked ethical dimension of AI documentation. Through a thematic analysis of 265 relevant documents, they identified six key themes, including model risks, appropriate use scenarios, and strategies to mitigate potential harms. Their research reveals a troubling gap: While developers often document technical specifications, they often overlook the ethical implications and risks associated with deploying AI systems. This oversight not only limits users' awareness of potential harm, but also undermines efforts to align AI deployment with societal values. Gao et al. emphasize that clear, thorough ethics documentation is critical to ensuring that AI systems are transparent, safe, and ethically sound. These practices, they argue, are essential to fostering trust and accountability in high-risk AI applications.

Together, these studies show that README files and similar documents are not just AIDS, but core components of responsible software and AI system development. They highlighted the role of detailed, transparent documentation in enabling effective collaboration, ensuring ethical compliance, and promoting user adoption. By addressing gaps in current practices - such as lack of consistency, inadequate ethical considerations, and incomplete data set descriptions - software projects can create a more inclusive and trusted development environment. A standardized framework for README documents can further enhance their value, making them an indispensable tool for aligning technological advances with user needs and societal expectations.

While recognizing the importance of README files, significant challenges remain in achieving consistency, integrity, and clarity. Both Liang et al. (2024) [3] and Yang et al. (2024) [5] highlight major gaps in document integrity and coverage of models and datasets. Liang et al. highlight that sections dealing with key aspects such as environmental impacts, limitations, and assessments are often incomplete, which reduces the utility of documentation to end users. Similarly, Yang et al. found that dataset documentation often lacked detailed descriptions of structure, licensing, and usage considerations, especially for less popular datasets. These gaps hinder users' ability to fully understand, evaluate, and effectively utilize software, creating barriers for both developers and end users.

In addition, Gao et al. (2024) [2]cite the lack of consistent ethical documentation as an urgent concern, especially for high-risk AI systems. They argue that without clear documentation of model risks, ethical use cases, and mitigation strategies, developers and users may inadvertently engage in unethical practices. Such inconsistencies not only limit trust in software, but also create potential risks for wider social applications. The study highlights the need for a standardized approach to documenting ethical considerations to ensure transparency and accountability.

These challenges point to a broader problem in the software engineering community: the lack of universally accepted standards for creating comprehensive and user-friendly documentation. Addressing these gaps is critical not only to improving availability, but also to fostering trust, promoting ethical practices, and ensuring responsible deployment of AI systems. By focusing on these areas, README files can evolve into powerful tools that bridge the gap between developers and users while supporting safe and efficient use of software.

8 Conclusion

Documentation plays a key role in the adoption and usability of machine learning (ML) models. A good readme can act as a gateway for potential users, making it easier for them to understand, implement, and trust the model. Our research highlights the importance of README files in model adoption and the significant impact that quality documentation can have on the wider machine learning community. This study aims to address gaps in README quality through a combination of best practices and automated tools, providing a pathway to more accessible and usable ML documents.

The README file is usually the first interaction a user has with an ML model. Clear, comprehensive documentation lowers the barrier to entry for new users, democratizes access to cutting-edge tools, and facilitates collaboration. Our findings confirm that sections like model descriptions and instructions for use are widely included, demonstrating their importance in the community. However, frequent omissions in sections such as model limitations and training details reveal gaps in transparency and usability that must be addressed. Without proper documentation, even the most advanced models run the risk of being underutilized. Users can only rely on trial and error or external sources to understand how the model works. In contrast, detailed README files reduce this friction, resulting in faster adoption and more efficient deployment of ML models in real-world scenarios.

Our analysis reveals the advantages and disadvantages of the current state of the README file. Most README files include basic sections such as descriptions (91%) and instructions for use (86%), confirming their foundational role in making the model easy to understand and usable. However, only 28% of README files address model limitations and only 35% provide training information. These omissions can lead to misuse of models, ethical issues, and user dissatisfaction. The absence of a section on restrictions is of particular concern. Users need to understand the boundaries of model performance to avoid deploying it in inappropriate scenarios. For example, omitting details about bias or accuracy in a given situation can lead to unintended consequences, such as exacerbating social inequality. Similarly, lack of training information limits users' ability to replicate results or fine-tune models for their specific needs. Our research highlights the need for README files to strike a balance between accessibility and technical depth. Descriptions and instructions for use ensure that a broad audience can participate in the model, while sections such as training details and restrictions cater to more advanced users who need transparency and replicability.

Improving the quality of readme documents is not without its challenges. One major obstacle is the unstructured nature of README content. Readme files are often written in different formats, which makes it difficult to apply consistent quality checks. Our work uses a Large language model (llm) to overcome this challenge, achieving an impressive 98% accuracy rate in identifying and evaluating README components. However, this approach requires careful fine-tuning and a lot of computing resources. Another challenge is the limited availability of high-quality labeled data for training our tools. Many README files lack a standardized structure, making it more difficult to extract meaningful insights. Despite these difficulties, our research shows that with targeted efforts, it is possible to build tools that effectively assess README quality.

The study has several important contributions. First, we created a best practice guide for writing a clear and comprehensive README file. The guide provides actionable recommendations for model developers, highlighting the inclusion of key sections such as descriptions, instructions for use, restrictions, and training details. Second, we developed a tool powered by the LL.M. to assess the quality of the README. The tool not only identifies missing pieces, but also provides feedback to improve the overall structure and content of the README file. By automating this process, we lower the barrier for developers to create high-quality documentation. Going forward, there are several ways to extend and improve this work:

- Use more data: Expanding the data set used for training evaluation tools will improve their accuracy and adaptability. Larger data sets will help the tool handle a wider range of README formats and content styles.
- Reduce computational costs: Exploring more efficient training methods or alternative models will make the tool easier to use for developers with limited resources.
- Automate README creation: Developing tools that automatically generate README files based on model metadata

and user prompts can significantly simplify document processing. This is especially valuable for smaller teams or open source contributors.

- Promote community standards: Working with the ML community to establish standardized README templates can improve consistency between projects. This will also make it easier for users to browse through documents for different models.

The importance of README files goes beyond technical availability. They are a key element in promoting transparency, ethical deployment, and collaboration in the machine learning community. High-quality documentation enables users to understand the capabilities and limitations of the model, thus ensuring responsible use. In addition, the comprehensive README file supports reproducibility, which is a cornerstone of scientific progress. By addressing gaps in README quality, this research contributes to the broader goal of making ML tools more accessible and trustworthy. The development of best practices and automation tools is a step toward standardizing and improving documentation across the field. As the field of machine learning continues to evolve, ensuring that models are accompanied by clear, transparent documentation is critical to maximizing their positive impact on society. In conclusion, improving README files isn't just about better usability, it's about fostering a culture of transparency, accountability, and collaboration in machine learning. This work lays the foundation for ongoing efforts to make ML models more accessible, ethical, and effective for a global audience.

References

[1] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology*, 3, 2, 77–101.

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

[3] Weixin Liang, Nazneen Rajani, Xinyu Yang, Ezinwanne Ozoani, Eric Wu, Yiqun Chen, Daniel Scott Smith, and James Zou. 2024. What's documented in ai? systematic analysis of 32k ai model cards. *arXiv preprint arXiv:2402.05160*.

[4] Johnny Saldaña. 2021. The coding manual for qualitative researchers.

[5] Xinyu Yang, Weixin Liang, and James Zou. 2024. Navigating dataset documentations in ai: a large-scale analysis of dataset cards on hugging face. *arXiv preprint arXiv:2401.13822*.