An Empirical Study on Hugging Face Trends, Topics and Challenges on Stack Overflow

Anonymous Authors

***Abstract*—Hugging Face (HF) has emerged as a pivotal platform for the Machine Learning (ML) community, functioning as a central hub where developers collaborate, share models, and exchange datasets. By offering a vast repository of pre-trained models (PTMs), HF has democratized access to advanced ML resources, promoting model reuse and accelerating the development of ML-based systems. Despite its rapid adoption in recent years, there remains a limited understanding of the challenges developers encounter when working with HF in general and PTMs in particular. Understanding these challenges is crucial for guiding future research and developing support strategies for the software engineering community. Consequently, in this study we investigate HF related Stack Overflow (SO) posts, one of the most popular discussion platforms for developers, to uncover the relevance of the topics, key challenges and trends in HF-related discussions. This understanding will help future studies and the HF community improve the use of HF by focusing on the challenges developers face according to the prevalence and complexity of each of these challenges. To do so, we apply a topic modeling technique, to categorize the topics discussed in SO posts that are related to HF. We then assess the popularity and difficulty of these topics to gain deeper insight into the specific challenges developers encounter. Our findings reveal an average annual growth rate of 31.3% in the number of HF-related questions on SO from 2019 to 2024. Furthermore, we identify eight major topics, with the usage and understanding of large language models (LLMs) being the most popular, while the distributed computing and resource management of PTMs stands out as the most challenging topic for developers.**

***Index Terms*—Hugging Face, Topic Modeling, Challenges, Stack Overflow.**

I. INTRODUCTION

THE rise of ML has not only transformed industries but also fostered the development of collaborative platforms that support open sharing of models, tools, and datasets. Among these, HF has emerged as a leading platform, widely recognized for its role in democratizing access to PTMs, deploying LLMs, and simplifying the complexities of ML development. HF provides a dynamic space where researchers and developers collaborate to build, refine, and deploy state-of-the-art models, making it a central hub in the ML and AI communities [1][2]. The platform offers a wide array of services, including model reuse, dataset access, and deployment solutions, which have catalyzed its rapid adoption globally.

Although prior research [2][3] has examined some of the challenges related to HF, comprehensive insights into the real-world issues faced by developers remain limited to a few issues and bugs. For instance, Jian *et al.* [2] interviewed 12 practitioners to better understand how they select pre-trained deep learning models (e.g., based on the popularity of models) from HF and challenges on the selection of these models. Their study identified a limited number of just three high-level challenges. Pan et al. [3] studied bugs related to the use of PTMs, while our study focus on challenges in general which might not be necessary source code bugs (e.g., integration of a model). On top of a lack of a deep understanding of challenges that are related to HF from developers perspective, there has been no exploration of the trends, topics, and challenges related to HF that arise on SO. Furthermore, these prior studies mainly focused on the PTMs of HF, while our study is on HF in general including its other features such as the datasets provided through HF. This gap in the literature limits our understanding of the common difficulties that developers face, and the key areas related to HF requiring further support or development. The growing body of SO discussions presents an opportunity to better understand the needs and pain points of the HF community.

In this study, we aim to enlarge our understanding of HF related challenges by conducting an empirical investigation into HF-related discussions on SO. Specifically, we explore the primary topics, their popularity, and the challenges developers face when using HF. We analyze the growth of these discussions over time with a focus on understanding how the community's engagement with HF evolves and what trends are emerging. By identifying the most pressing issues, we aim to guide future research and support strategies for the HF community. To achieve these objectives, we apply well-established topic modeling techniques to categorize the extensive HF-related discussions on SO. Our quantitative and qualitative analyses provide an understanding of developers interactions, the most common issues they encounter, and how these issues have evolved over time.

Notably, our study captures a sharp increase in user interest and engagement with HF, underscoring the platform's growing importance in the ML landscape. More specifically, our analysis reveals 4,744 questions and 3,341 answers related to HF on SO, contributed by 5,838 distinct users between 2019 and 2024. The number of HF-related questions has grown significantly, from just 15 in 2019 to over 4,744 by 2024. However, 75.7% of these questions remain unresolved (i.e., with no accepted answers), highlighting the difficulties developers face in navigating HF's tools and models. Additionally, we uncovered eight key topics. The most prevalent is the usage and understanding of large language models (LLMs), which dominates discussions due to the increasing reliance on LLMs in natural language processing tasks. Within this topic, subcategories such as embeddings and tokenization are frequently discussed, as they are crucial to the performance

and accuracy of language models. However, developers also encounter significant challenges, particularly with distributed computing and resource management for PTMs. This topic is considered the most difficult because it involves managing large models across multiple nodes, optimizing resource allocation, and maintaining efficiency in environments with limited computational capacity. In contrast, topics such as model deployment, though still important, appear to pose fewer technical difficulties, probably because the deployment process is better understood and supported by existing frameworks. The main contributions of this study are summarized as follows:

- We conduct an empirical study to systematically mine and analyze HF-related questions on SO, identifying key trends, topics, and challenges.
- We apply a topic modeling technique to uncover the core topics discussed within the HF-related posts, providing insights into the common challenges developers face.
- We publicly provide a replication package[1] that includes the dataset and scripts, allowing other researchers to replicate and build on our findings.

The remainder of this paper is structured as follows. Section II details our approach. Section III presents our empirical study, including research questions and empirical results. Section IV discusses the implications of our findings. Section V reviews related work. Section VI addresses the threats to the validity of our study, and Section VII concludes our work.

II. APPROACH

The objective of our study is to investigate the trends within HF discussions, such as user engagement, the growth of questions, and the evolution of tags over the years, with a primary focus on understanding the challenges developers face when working with HF. To achieve this, we analyzed discussions on SO related to HF, relying on a topic modeling technique to gain a deeper understanding of the topics, challenges, and experiences developers encounter when using HF. Figure 1 provides an overview of our study design, which we detail in following.

Step 1: identify HF-related tags. Every post on SO is accompanied by tags that provide a hint related to the topic of the post. To identify the most relevant HF-tags, we used the SO search tool within the tag section with the keyword "*huggingface*". This approach allowed us to retrieve the most relevant tags associated with HF including "*huggingface-transformers*", "*huggingface-tokenizer*", "*huggingface-datasets*" and "*huggingface-trainer*" which reflect key components of the HF ecosystem.

Step 2: Query & download SO posts. To retrieve the relevant posts, we leveraged the Stack Exchange API [4], a tool widely used by developers for data collection and analysis on Stack Exchange sites such as SO, Super User, and Ask Ubuntu. We considered a Python script using the `requests` library, making GET requests to the API endpoint with the appropriate parameters. This approach allowed us to collect all SO questions tagged with the identified HF-related terms. We then saved the data in a CSV file, resulting in a dataset of 4,744 questions.

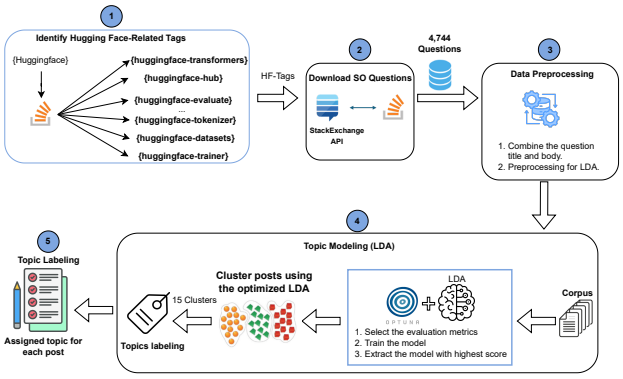

Fig. 1: Overview of our Approach

Step 3: Data Preprocessing. Before applying topic modeling, we preprocessed the dataset by removing irrelevant information. We stripped out code snippets and HTML tags using `BeautifulSoup` [5] for HTML parsing and the `re` [6] module for regular expressions. We then removed numbers, punctuation, common stop words, and frequent words such as *question* and *answer*. Using `Natural Language Toolkit` [7], we performed *lemmatization*, and *stemming* to standardize word forms. Additionally, we built a bigram model with `Gensim` [8] to capture frequent word pairs (e.g., "*inference endpoint*"), treating them as single units for topic modeling. This resulted in a clean, preprocessed dataset ready for analysis.

Step 4: Topic Modeling. To group similar posts in the same cluster and create the topic distribution in our corpus, we used of `Latent Dirichlet Allocation (LDA)` [9] model, a state-of-the-art topic modeling technique that has been widely used in previous studies [10][11][12][13]. LDA assigns a probability to each question for each topic, showing how likely the question is to be related to that topic. The topic with the highest probability is considered as the question's dominant topic.

***Hyperparameter Tuning*.** Achieving optimal outcomes with LDA necessitates effective hyperparameter tuning. The primary parameter required for LDA is the estimated `number of topics` to be generated. Opting for a lower value may yield broad topics, while a higher value could generate a lengthy list of detailed and non-informative topics. Moreover, LDA's performance significantly relies on the `chunk size`, the number of `passes` as well as other hyperparameters such as `eta`, and `alpha`. Specifically, the `chunk size` determines how many documents are processed in one batch, while the `passes` controls how many times the LDA algorithm will go through the entire corpus. `Eta` influences how words are distributed across topics. A lower `eta` results in topics dominated by fewer words, making topics more distinct, while

[1]https://anonymous.4open.science/r/Replication-Package-HFproject-1E71/

a higher `eta` allows for more words to be relevant to multiple topics, potentially leading to overlapping or less distinct topics. Finally, the `alpha` hyperparameter controls the distribution of topics within documents. A lower `alpha` value promotes the focus on fewer topics, while a higher `alpha` promotes a more even distribution of the documents across multiple topics.

Identifying the optimal LDA configuration poses a considerable challenge, resembling a combinatorial problem due to the vast search space available. Thus, we use a hyperparameter optimization framework, `Optuna` [14], to efficiently search the hyperparameter space and find the optimal combination of hyperparameters. We have chosen *Optuna* as our hyperparameter optimization framework for several reasons. First, `Optuna` supports many search algorithms such as Grid search, Random Search and NSGA-II [14]. Second, it simplifies the implementation process of hyperparameter tuning by requiring only the specification of hyperparameter search spaces along with fitness functions. Furthermore, `Optuna` provides a real-time dashboard with visualization tools, which is invaluable to monitor the optimization history, understand the importance of different hyperparameters, and ensure efficient convergence toward the optimal configuration.

Population & Individuals. In our study we selected NSGA-II as a search algorithm for `Optuna` since it is widely used in the literature and showed its performance to solve many search-based optimization problems [11][15][16]. In our context, each individual (also called a solution) represents a configuration of the LDA model's hyperparameters. In each generation, NSGA-II creates a population that contains a fixed number of individuals. NSGA-II iteratively optimizes the generated population (i.e, improving the fitness value of the individuals) by applying a set of genetic operators [17].

Fitness Functions. One of the popular metrics for evaluating a topic modeling task is the coherence score CV. It measures the consistency of words in a given topic to evaluate the interpretability and meaningfulness of a topic by computing the level of semantic similarity among words that are included in the topic [18]. At the beginning of our experiments, we relied solely on the coherence metric to guide the optimization process. As a result, our LDA model provided coherent topics, but we noticed redundancy in the word distributions among topics. We therefore decided to add another metric to our optimization criteria. Specifically we used the *Jensen-Shannon Divergence* (JSD) [19] to measure the diversity of the identified topics. This metric measures the inter-topic distance between all distinct pairs of topics. The greater the JSD inter-topic distance, the higher the diversity of the topics.

Optimal Solution. At the end of the genetic search, we obtain a set of individuals that form the Pareto fronts [17] of the genetic search. Such front contains solutions that offers diverse trade-offs between the fitness functions and better optimize the search process. We employ the *Knee point* technique [20] to select the optimal solution from the Pareto front, as suggested by a number of previous studies [21][22]. To determine the *Knee point*, we first extract the ideal point, where the coordinates represent the highest fitness scores (CV_{max},JSD_{max}) obtained from all solutions in the Pareto front, with each fitness function evaluated independently. Giving the solutions at the Pareto front P=(S_i,...,S_p), the *Knee point* $S_k \in P$ corresponds to the solution that minimizes the distance $\sqrt{(CV_{max} - CV(S_i))^2 + (JSD_{max} - JSD(S_i))^2}$ for all $S_i \in P$ [20]. As a result of tuning our LDA model using `Optuna` under 500 trials, we found that the optimal solution that maximizes the diversity and coherence scores yield to setting the `number of topics` to 15, `passes` to 250, `iterations` to 4.390, `eta` to 0.01, `alpha` to 0.76, and `chunk size` to 50. This configuration produced a model with diversity score of 0.65 and coherence score of 0.4.

Step 5: Topic Labeling. The LDA model provides a Document-Topic Distribution, showing how likely each question belongs to a specific topic, resulting in clusters of questions. Each cluster represents a topic, with questions most strongly associated with that topic. Additionally, the Topic-Word Distribution shows the likelihood of each word appearing in a topic, as shown in Table I. To assign labels to our topics, we followed the methodology outlined in prior research [23][24]. Initially, each author independently proposed labels based on the topic-word distribution. These preliminary labels were refined through iterative meetings until we reached substantial agreement (kappa=0.70). For example, the words in *Topic 2* (train, finetun, data, etc.) led us to label it "*Model Training*". To verify and refine labels, we sampled 50 questions from each cluster, focusing on those with the highest probability of belonging to their cluster. If the initial label accurately reflected the cluster, we kept it,otherwise, we adjusted it. We achieved substantial agreement (kappa=0.65) in the validation process. Detailed descriptions of each topic follow in the next section.

Topic	Word distribution
1	bert, label, model, class, layer, classif, implement, pytorch, size, shape
2	train, finetun, data, custom, trainer, want, tutori, fine_tun, format, creat
3	file, download, imag, script, local, deploy, creat, folder, directori, contain
4	transform, python, version, librari, instal, import, pytorch, tensorflow, work, packag
5	dataset, data, evalu, column, step, process, epoch, like, accuraci, metric
6	model, load, pretrain, save, weight, checkpoint, initi, like, config, ab
7	time, memori, infer, run, process, set, chang, batch_siz, take, valid
8	**8.1)** sentenc, transform, pipelin, encod, convert, list, emb, embed, possibl, similar
	8.2) token, word, like, probabl, split, give, roberta, vocabulari, want, add
	8.3) output, text, input, sequenc, length, batch, gener, decod, long, pad
	8.3) differ, predict, model, loss, task, languag, test, result, mask, score
Others	function, return, method, valu, give, paramet, expect, pass, tensor, wrong
	need, document, help, sure, creat, suggest, link, understand, build, read
	gener, know, base, current, specif, want, show, context, inform, prompt
	code, get, run, work, line, issu, notebook, messag, problem, google_colab

TABLE I: Topics and keywords

III. EMPIRICAL STUDY

This section describes the empirical study, including the research questions we address, our experiments, and results.

RQ1. (Trends) How have the discussions about HF evolved on SO?

Motivation. This research aims to examine whether HF-related topics on SO are still trending or if interest is declining, which may suggest that HF is becoming less relevant for ongoing study on the platform.

Approach. To address this research question, we first analyze the volume of HF-related questions and answers over time. Next, we examine trends in community engagement with HF topics, exploring how participation has evolved. Additionally, we track the annual involvement growth of HF users on SO. Finally, we extract and analyze the tags associated with HF-related questions, investigating how their usage has evolved over time.

Results. In the following, we describe the trends associated with HF discussions on SO.

- Trends of Posts (question & answer). Figure 2 highlights the annual count of HF-related questions and shows the yearly distribution of questions with and without accepted answers. The upward trend reflects an increasing interest in or adoption of HF models and technologies. The most notable finding is the sharp increase in the volume of questions from 2022 to 2023, where there was a significant 106% increase in this period. This could suggest either a surge in HF users or expanding use cases and challenges related to HF technologies. However, the decrease in the number of questions in 2024 can likely be attributed to the fact that the data represent only a partial year, which naturally results in fewer questions compared to a full year such as 2023. Additionally, as the HF platform matures, improved documentation and resources may have reduced the need for users to ask questions on SO. On the other hand, an accepted answer represents a solution chosen by the question asker, signaling that their inquiry has been resolved. However, the figure reveals a concerning trend, while the total number of questions is increasing, the gap between questions with accepted answers and those without is widening. Specifically, a growing number of questions remains unanswered or lack an accepted solution, indicating that more users are struggling to find satisfactory resolutions.

This trend is further supported by data in Table II, which shows that a significant 75.7% (1,153 out of 4,744) of questions do not have an accepted answer. Furthermore, we found that the average number of answers per questions is only one. This high percentage of unresolved questions and very low response rate could be indicative of the challenges involved in working with HF technologies. This also suggests that the community is struggling to keep pace with the rapid expansion and diverse challenges that users are facing. Such findings point to potential areas of improvement in community support and resources for HF users.

- Trends of Users. We also investigated the involvement of the SO users in HF discussions. This includes analyzing the trends of unique users who post questions and answers over the years. Figure 3 shows a general growing interest in HF, which reflects its rising popularity. The significant jump in the number of users, especially from 2022 to 2023, suggests that HF may have introduced new features, improvements, or gained wider recognition in the HF community during this period. This pattern is similar to the number of questions asked, with an upward trend between 2019 and 2023, but again, a decline in 2024.

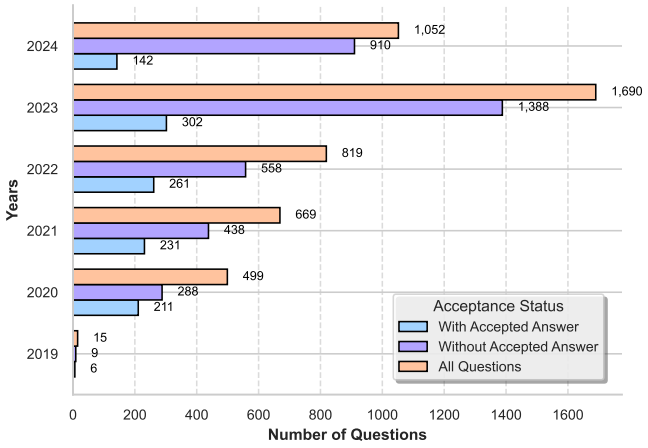

Fig. 2: Yearly Distribution of Questions With and Without Accepted Answers

TABLE II: Statistics about the collected data

Item	Count
Number of posts	8,085
Number of Questions	4,744
Number of answers	3,341
Number of answered questions	2,618
Number of accepted answers	1,153
Number of distinct tags	890
Number of distinct users	5,838
Average number of tags per question	4
Average number of answers per question	1

TABLE III: Top-10 Most Popular Tags with Percentage of Average Yearly Increase

Tag name	Tag count	Percentage of Average Yearly Increase
Huggingface-Transformers	3,800 (22.6%)	135.4%
Python	2,097 (12.5%)	82.6%
Pytorch	1,242 (7.4%)	43.4%
NLP	1,100 (6.5%)	30.8%
Bert-Language-Model	636 (3.8%)	8.4%
Huggingface-Tokenizers	563 (3.3%)	16.8%
Machine-Learning	377 (2.2%)	18.6%
Large-language-model	360 (2.1%)	37.8%
Deep-Learning	329 (2.0%)	12.8%
Huggingface-dataset	318 (1.9%)	12.2%

- Trends of Tags. We identified 890 unique tags associated with HF-related questions. Table III shows the ten most popular tags. The "*Huggingface-Transformers*" tag (22.6%) is the most frequently used and has demonstrated significant annual increase (135.4%), reflecting its central role in the HF

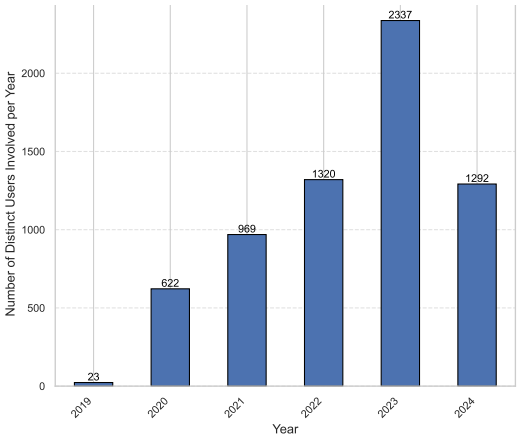

Fig. 3: Distinct Users Involved per Year

ecosystem. Similarly, "*Python*" (12.5%) and "*PyTorch*" (7.4%) are among the most common and rapidly growing tags with an average annual increase of 82.6% and 43.4%, respectively. This underscores Python's importance in machine learning and NLP, with PyTorch emerging as a leading deep learning framework within HF, aligning with findings in [25]. These top tags highlight key tools, frameworks, and techniques in HF-related discussions, with consistent annual increases that indicate increased user engagement in SO.

Answer to RQ1

Our analysis reveals a notable increase in discussion activity on HF starting from 2022, with contributions from over 5,838 unique users asking a total of 4,744 questions. Despite this rise, 75.7% of the questions remain without an accepted answer, indicating that many users still struggle to find adequate solutions. Our tag analysis, based on 890 distinct tags, shows that "*Huggingface-Transformers*" (22.6%) is the most frequently used tag, highlighting its prominent role in the HF ecosystem.

RQ2. (Topics) What are the key topics discussed about HF?

Motivation. We aim in this research question to understand the topics discussed about HF on SO. Understanding these topics is crucial for gaining insights into the areas of interest and concerns within the community. Such analysis can help identify the most common challenges encountered by developers when working with HF.

Approach. To successfully identify the discussed topics related to HF on SO, we employed the topic modeling-based approach using LDA, as described in Section II.

Results. As depicted in Table I, the LDA clustering identified 15 clusters associated with HF questions on SO. Four of these clusters lack a dominant topic and are classified as "*Other*". Among the remaining 11 clusters, four contain questions related to the usage and understanding of Large Language Models (LLMs). Consequently, we unify these clusters under a main topic with three subtopics. Ultimately, we obtained the following eight topics:

- Topic 1: Model Customization
- Topic 2: Model Training
- Topic 3: Model Deployment
- Topic 4: Environment
- Topic 5: Datasets
- Topic 6: Model Loading, Saving, and Pushing
- Topic 7: Distributed Computing and Resource Management
- Topic 8: LLMs Usage and Understanding
  - Sub-topic 8.1: Embeddings
  - Sub-topic 8.2: Tokenization
  - Sub-topic 8.3: Incomprehensible outputs from LLMs

These topics are also presented in Table I where we show for each one a partial set of the Topic-Word Distribution. In the following, we introduce each topic with a brief description and a sample of related questions on SO.

Model Customization: It involves applying changes to the model to tailor it to specific needs, including architectural changes, and integration with other models. In this topic, we observe that users attempt to understand or modify the architecture of models from the HF transformers library but run into various challenges. Such challenges generally arise when adding new layers to PTMs, removing existing layers, or if a shape mismatch exists between the model's logits and the labels of the inputs. Quote 1[26] shows an example of this particular challenge where the user is seeking an explanation for a shape mismatch error between logits and labels when integrating a transformer model with a custom model.

***Quote 1*: ValueError - Shape mismatch**

💬 *"[...] I am trying to use a transformer model as an embedding layer and feed the data to a custom model.[...] how can I solve this? and why is the (10, 30) shape required? [...]"*

Model Training: HF provides a wide range of PTMs and offers a `Trainer API` designed to simplify the training and fine-tuning of transformer models. The `Trainer API` delivers a comprehensive training interface in PyTorch, supporting distributed training across multiple GPUs/TPUs. Users frequently encounter challenges related to training transformer models using the `Trainer API`. These challenges often arise when configuring the trainer object with its various arguments or when attempting to allocate the appropriate computational resources for the HF trainer. A common challenge in Quote Quote 2[27] involves calculating and visualizing evaluation metrics, such as the confusion matrix, precision, recall, and F1-score, after fine-tuning a model using the HF `Trainer API`.

***Quote 2*: How can I check a confusion_matrix after fine-tuning with custom datasets?**

💬 *"[...] After finishing the fine-tune with Trainer, how can I check a confusion_matrix in this case?"*

Model Deployment: it refers to the process of exposing the model inference to the public. HF offers a service called `Spaces` [28] that allows users to deploy, share, and showcase model demos in a user-friendly interface. Deploying a model in production usually requires performing a series of tasks including packaging the model in a container, provisioning the infrastructure, and more. HF offers a second service called `Inference Endpoints` [29] that allows users to deploy ML models directly from the HF hub to managed infrastructure on their favorite cloud service provider including *Amazon SageMaker* and *Azure ML*. In this topic, users are mainly facing challenges when dealing with the above-mentioned services. Challenges are mainly about the deployment environment (e.g., a missing library), or creating HF space using various frameworks such as *Gradio*, *FastAPI*, or *Flask*. For instance, in Quote 3[30] the user is failing to deploy his model to a *SageMaker* endpoint due to a missing library.

***Quote 3*: Invoking SageMaker endpoint failed since timm library was not found in the environment**

"I deployed a Hugging Face model [...] but it seems like this package is missing from the SageMaker endpoint. How can I add it as part of the model deployment?"

Environment: An environment refers to the specific setup that includes all the necessary tools, libraries, and dependencies required to run a program. Users frequently face challenges related to environmental issues when executing their code. Common errors include "*ImportError*" or "*ModuleNotFoundError*", which are often due to missing modules, compatibility issues, or dependency conflicts. Quote 4[31] shows an example in which a user faces an issue involving dependency conflicts. Specifically, the problem is attributed to compatibility issues between the used packages and the user's environment, which includes Python and Rust compiler versions.

***Quote 4*: Installation error with pip install tokenizers==0.12.1 – Compatibility issue with Python 3.6.15 and Rust 1.72.0**

"I am encountering an issue while trying to install the tokenizers package with version 0.12.1 using the command pip install tokenizers==0.12.1. My Python version is 3.6.15, and the Rust compiler version is 1.72.0."

Datasets: In addition to PTMs, HF provides access to over 100,000 ready-to-use datasets through its `Datasets` library [32]. This library includes the `load_dataset` method, which allows users to conveniently load datasets either from the HF hub or from a local disk. These datasets can then be used for a variety of tasks, such as model training, and data preprocessing. In the case where the dataset is too large which can introduce memory issues, HF allows loading the dataset as streamed data using the `IterableDataset` object. In this topic, users often encounter challenges with various aspects of dataset handling, including loading datasets, pushing datasets to the HF hub, and performing conversion operations. For instance, converting an `IterableDataset` to a static dataset or transforming a Pandas DataFrame into a HF dataset can be complex. For example, in Quote 5[33] the user is working with a large dataset and is using an "*IterableDataset*" to handle it. However, the user encounters difficulties when attempting to save the dataset locally after performing certain operations, as the "*IterableDataset*" does not have a *save_to_disk()* method. The user is looking for a way to convert the "*IterableDataset*" into a regular dataset that can be stored in memory or on disk.

***Quote 5*: Can I convert an IterableDataset to Dataset?**

"I want to load a large dataset, apply some transformations[...] and store as files so I can later on just load from there.[...] but then I would expect some way to convert an iterable to a regular dataset (by iterating over it all and store in memory/disk, nothing too fancy)."

Model Loading, Saving, and Pushing: Model loading in HF is a key feature that allows quickly and efficiently use PTMs for a variety of tasks. The HF transformers library provides an interface for loading these models [34]. Additionally, HF users can push their models to the HF Model Hub, making them available for others to use, adapt, and build upon in their own projects. Moreover, users can save their trained models in a way that facilitates reuse, sharing, and deployment in various applications. In this topic, when performing the aforementioned operation, users are facing multiple challenges. Such challenges are mainly about weights initialization, memory errors when loading models, or getting inconsistent prediction when reloading the model. Another example is shown in Quote 6[35], where issues are encountered with the conversion of a model's format when pushing it to the HF hub. Specifically, a model saved in the *.safetensors* format is automatically converted to the .bin format during the upload process, leading to difficulties in preserving the original format.

***Quote 6*: Why does Hugging Face's push_to_hub convert saved models to .bin instead of using safetensor mode?**

"I am attempting to push a saved model in model-00001-of-00006.safetensors mode, but the model gets converted to pytorch_model-00001-of-00006.bin before being saved to the hub. How can I prevent this?"

Distributed Computing and Resource Management: This has significant implications for the training, inference, and deployment of machine learning models. Advanced architectures like transformer models (e.g., BERT, GPT) demand substantial computational power for both training and inference, making efficient resource utilization a key concern. HF offers the `Accelerate` library [36], a powerful tool that enables PyTorch code to run seamlessly across various distributed configurations with minimal changes requiring just four lines of code [37]. In this topic, we found that the primary challenges developers face relate to the effective use of the HF `Accelerate` library and managing resource allocation during training or inference. These challenges often involve configuring the `Accelerate` library, such as selecting appropriate parallelism techniques for distributed setups. For

instance, in Quote 7[38], the user is experiencing a device mismatch error when calculating metrics on a multi-GPU setup using Accelerate. Output tensors are on different devices than input tensors after a forward pass, causing an error. The user is therefore seeking advice on resolving this device inconsistency.

***Quote 7*: HuggingFace accelerate device error when running evaluation**

"I am running some experiments on a multi-GPU cluster, and I am using accelerate.[...]. While the training code seems to work fine using accelerate (it utilizes multiple GPUs), I run into an error when trying to calculate said metrics. It seems that after doing a forward pass when evaluating the output tensors are put on another device than the input tensors."

LLMs Usage and Understanding: Large Language Models have drawn a lot of attention due to their strong performance on a wide range of natural language tasks. LLMs' ability of general-purpose language understanding and generation is acquired by training billions of model's parameters on massive amounts of text data [39]. HF offers various pretrained LLMs including GPT-2, T5, Bert, Bart and others. These models can be used for various applications such as text generation, translation, summarization, and more. We found that questions related to LLMs on HF typically fall into three main subtopics: (1) Embeddings, (2) Tokenization, and (3) Incomprehensible outputs from LLMs. Those subtopics are described in the following.

1) Embeddings: Embedding is the process of converting entities (i.e., sentences in general) into vector representations, in a way that captures their semantic relationships [40]. In this topic, we observe that most of the questions are procedural questions (i.e., How-questions). This indicates that users primarily struggle with implementing and following specific processes rather than understanding the underlying concepts. Specifically, users are encountering challenges in extracting embedding while using HF models. Such challenges arise in general when (1) converting embeddings back to text, (2) addressing positional embedding (i.e., embedding that learns the meaning of word position in a sentence) [41], or (3) when using the HF pipeline method that allows extracting embedding using a specific PTM and tokenizer. We present an example of such challenges in Quote 8[42], where a user is exploring the possibility of ignoring positional embeddings when using BERT. Positional embeddings are a critical component of transformer models like BERT, as they enable the model to capture the order of words in a sequence. However, exploring the removal of positional embeddings could provide insights into how much BERT's performance relies on these embeddings for certain use cases.

***Quote 8*: BERT without positional embeddings.**

" I am trying to build a pipeline in HuggingFace which will not use the positional embeddings in BERT, in order to study the role of the embeddings for a particular use case. [...] Will I need to modify BERT source code, or is there a configuration I can fiddle around with?"

2) Tokenization: In natural language processing, tokenization refers to the process of converting a sequence of text into smaller parts, known as tokens [43]. HF offers a variety of tokenizers with different tokenization methods, each tailored to different types of models and tasks. In this topic, we found that users are mostly inquiring about the tokenizer vocabulary, specifically looking at how to extend a vocabulary, and remove an existing token. Quote 9[44] shows an example where the user is trying to extend and customize the tokenization process to handle specific words or terms relevant to their domain that may not be covered by the standard tokenizer.

***Quote 9*: Replace special tokens in a tokenizer to add domain-specific words in BERT based models - Hugging Face**

"Let's say I have domain-specific word that I want to add to the tokenizer [...]. However, I have not found an example or documentation that shows how to achieve that."

3) Incomprehensible Outputs from LLMs: This subtopic addresses issues users face when the outputs from LLMs are difficult to interpret or justify. Problems often include scenarios where the generated text is identical to the input, or when the output is consistently too short. Another example is illustrated in Quote 10[45], where the user attempts to perform a masked language modeling task using the T5 model. In this task, masked tokens in the input sequence are expected to be replaced by the model's predicted tokens. However, instead of providing a valid token for the masked segment, the T5 model generates an unexpected or incorrect output.

***Quote 10*: How to use output from T5 model to replace masked tokens in input sequence**

"I am working with the T5 model from the Hugging Face Transformers library and I have an input sequence with masked tokens that I want to replace with the output generated by the model. Here is the code."

Answer to RQ2

Our qualitative analysis identified eight primary topics discussed by HF users on SO. These topics include: (1) Model Customization, (2) Model Training, (3) Model Deployment, (4) Environment, (5) Datasets, (6) Model Loading, Saving, and Pushing, (7) Distributed Computing and Resource Management, and (8) LLMs Usage and Understanding.

RQ3. Which HF-related topics are most popular and difficult?

Motivation. This research aims to identify the most popular and challenging HF-related topics on SO to understand community pain points and areas needing more support, helping to address the needs of developers and researchers in HF technologies.

Approach. To analyze the popularity and difficulty of HF-related topics, we use six key metrics. Similar to prior works [10][46][23], we consider three popularity metrics that measure (1) the number of questions per topic, (2) the average

TABLE IV: Popularity and Difficulty of HF Topics

Topic	Popularity Metrics			Difficulty Metrics		
	#Questions	Avg. Views	Avg. Voting Score	% without accepted answer	% without answers	Median time (days)
LLMs Usage and Understanding	**1238 (26.1%)**	1286.4	**1.4**	73.0	42.4	13.2
Model Customization	572 (12.1%)	900.3	1.0	74.7	45.6	6.0
Environment	467 (9.8%)	2133.5	1.2	76.4	44.5	12.7
Model Loading, Saving, pushing	421 (8.9%)	**2193.7**	**1.4**	76.7	42.8	19.2
Model Deployment	315 (6.6%)	1748.5	1.3	79.4	46.0	34.0
Model Training	311 (6.6%)	770.9	1.0	81.0	**51.0**	15.5
Distributed Computing and Resource Management	297 (6.3%)	1040.7	1.1	**82.2**	**51.0**	**37.4**
Datasets	251 (6.5%)	875.7	0.9	77.7	44.2	19.3

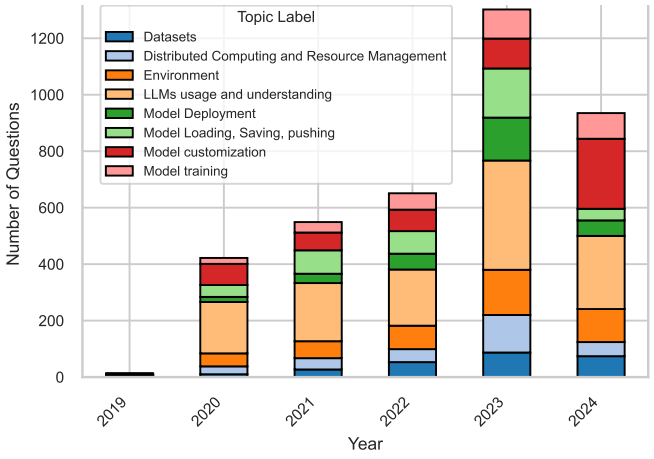

Fig. 4: Annual Distribution of Questions Across HF Topics

number of views (avg. views) per topic, and (3) the average voting score (avg. scores) per topic. Additionally, we use three metrics, previously established in related works [47][24][48], to evaluate the difficulty level of each topic. These metrics include (1) the number of unanswered questions per topic, (2) the number of questions without an accepted answer per topic, and (3) the median response time per topic.

Results. Table IV shows the topics popularity and difficulty results.

Popularity. Our analysis indicates that "*LLMs Usage and Understanding*" is the most popular topic based on the number of questions and average voting score. Our results show that 26% of HF-related questions pertain to "*LLMs Usage and Understanding*" topic, being the most dominant topic. On the other hand, while the topics "*Environment*" and "*Model Loading, Saving, and pushing*" are more popular in terms of average views, they have a significantly lower number of questions (421 questions, accounting for just 30% of the total questions related to "*LLMs Usage and Understanding*"). This pattern suggests that although these topics exhibit substantial interest, as evidenced by the high view count, the fewer questions may suggest that these topics are well-answered, allowing users to find answers without asking new questions.

Finally, when examining the popularity of topics over the years (see Figure 4), "*LLMs Usage and Understanding*" has consistently remained the most popular topic in terms of the number of questions, from 2020 to 2024. This topic has shown an average annual increase of 17.8%, underscoring its growing relevance and dominance in discussions over time.

Difficulty. We observe that the topic of "*Distributed Computing and Resource Management*" is the most difficult in terms of the number of questions without accepted answers (82.2%), unanswered question (51.1%), and the median time for questions to receive accepted answers (37.4 days) as shown in Table IV. In this topic, the large gap between unanswered questions and unaccepted answers suggests that while there is some level of engagement (since almost 50% of questions receive at least one answer), the quality or applicability of those answers may be lacking. This indicates that this topic is complex and that the community is struggling to provide definitive solutions. The topic "*Model Training*", despite being second in difficulty in terms of number of questions without accepted answer, the median time to receive an accepted answer is only 15.5 days, suggesting that when an answer is given, it is resolved relatively quickly. On the other hand, the topic "*LLMs Usage and Understanding*", despite its popularity, the percentage of questions without answers (42.4%) and without accepted answers (73.0%) is relatively moderate compared to more difficult topics. This suggests that while many users are actively discussing and engaging with this topic, the community is still able to provide answers or solutions at a reasonable rate. This might indicate that, although it is popular, the challenges related to LLMs are manageable with community support.

Answer to RQ3

Our analysis reveals that *"LLMs Usage and Understanding"* is the most popular topic based both on the number of questions and the average score. However, *"Environment"* and *"Model Loading, Saving, and Pushing"* have higher average views but significantly fewer questions, indicating that these topics are likely easier to address. In contrast, *"Distributed Computing and Resource Management"* stands out as the most difficult topic.

IV. DISCUSSION AND RECOMMENDATIONS

This study provides insights into the trends and challenges associated with HF usage as observed through discussions on SO. By analyzing a large dataset of posts from 2019 to 2024, we identified eight core topics of discussion, with "*LLMs Usage and Understanding*" emerging as the most prominent. The increasing number of HF-related questions highlights the platform's growing importance in the ML community. However, the high percentage of unanswered questions (44.8%) and those without accepted answers (75.7%) suggests that despite its popularity, developers continue to encounter significant hurdles. In the following, we will describe how our findings can provide valuable insights to the HF community, including practitioners and researchers.

A. Implications for Practitioners

Increased Community Engagement: The marked increase in discussions from 2022 (RQ1) signals the growing need for practitioners to solve technical challenges when using HF. This reflects the rapid adoption of HF technologies, suggesting that there is a continuous demand for accessible support and solutions on platforms such as SO.

Documentation Improvement: Through our analysis, we observed that most of the questions (23.6%, 1,122 out of 4,744) are "*how*" questions where users are generally asking about a method or technique to implement an element related to HF technologies. Questions of this type differ from the "*why*" questions (3.6%, 170 out of 4,744) as here the user has a particular goal in mind and asks for the steps to achieve this goal (e.g.,"*How to compare sentence similarities using BERT embeddings*"). Therefore, this high number of "*how questions*" suggests that HF provides dedicated documentation and more examples for specific tasks. We also observed cases in which practitioners follow existing tutorials or read the HF documentation but do not get their expected results (e.g.,"*I am trying to fine-tune a RoBERTa model for the first time following this tutorial [..]. However, I am obtaining this error [..]*"). Including a section in the documentation on common pitfalls and real-world use cases, along with encouraging community contributions for sharing experiences and solutions, would greatly enhance its value.

B. Implications for Researchers

In-Depth study on LLMs usage with HF: Since LLM usage and understanding became the most popular topic (RQ3), we recommend conducting an in-depth study to explore specific challenges and advancements in this area. For instance, delving into some challenging subtopics such as *"Tokenization"*,*"Embeddings"*, and working on how to improve *"Incomprehensible outputs from LLMs"* could significantly contribute to developing improved methodologies, best practices, and tools for users working with LLMs.

Future Research Directions for Efficient Resource Management in Distributed Computing: To address the significant challenges developers face with HF's accelerate library (RQ3), future research should prioritize creating more adaptive and efficient resource management solutions. Researchers should focus on designing solutions to support parallelism and memory allocation algorithms that automatically optimize settings based on model size and hardware capacity, for example. Furthermore, exploring innovative approaches to automated distributed setup configurations, especially for resource-limited environments, could significantly improve the usability of large-scale models.

V. RELATED WORK

Prior studies on HF are centered on distinct elements of the HF ecosystem, such as addressing security vulnerabilities, examining environmental impacts, and exploring the challenges involved in reusing PTMs.

For example, Jiang *et al.* [2] conducted an interview with practitioners using HF and identified various challenges in reusing PTMs. They extended their findings with a security risk analysis based on information from the HF Hub, concluding that the supply chain of PTMs has several risky practices, such as the frequent absence of signatures (i.e., cryptographic signatures that ensure the integrity and authenticity of models). Taking a security perspective, Kathikar *et al.* [49] analyzed the linked GitHub repositories of 110,000 HF models using static analysis. They identified numerous vulnerabilities, most of which were of low severity. However, a higher proportion of high-severity vulnerabilities were found in popular fundamental repositories such as Transformers, complicating the task of securing ML models.

To explore the impact of HF on environmental sustainability, Castaño *et al.* [1] examined around 170K models and discovered that only a few repositories provided data on the carbon emissions during model training. Then, in a following study [25] they investigated over 380,000 models on HF using data from the HF Hub API and the *HFCommunity* dataset to explore community engagement, evolution, and maintenance. They evaluated model maintenance, classified commit messages, and introduced a system to estimate maintenance status. They showed that highly maintained models tend to be more popular, larger, and better documented than their less maintained counterparts. Ait *et al.* [50] aimed to streamline the analysis of HF data and developed *HFCommunity*, a tool that aggregates and integrates data from the HF Hub, including repositories, discussions, files, commits, and more.

McMillan *et al.* [51] proposed creating documentation for language datasets and natural language generation models, leveraging resources from the HF Hub and the GEM (Generation, Evaluation, and Metrics) benchmark which is made for natural language generation models. Pan *et al.* [3] investigated GitHub repositories of popular models supported by HF Transformers. They analyzed reported bugs to develop a detailed taxonomy, identifying root causes and implications for model reuse. Taraghi *et al.* [52] explored the challenges, benefits and trends of reusing PTMs on HF mainly using HF Forums and not SO. Through qualitative and quantitative analyses, the authors identified key issues such as limited guidance

for beginners, difficulties in understanding model outputs, and insufficient documentation. The study compared model types discussed on the HF Forums with those on the hub, finding BERT as the most discussed and uploaded model. It also revealed a negative correlation between model availability and the number of related discussions.

Previous studies have predominantly concentrated on distinct aspects of the HF ecosystem, such as security vulnerabilities, environmental impacts, and the challenges surrounding the reuse of PTMs. In contrast, our work adopts a more expansive approach, encompassing a broader scope of objectives. Instead of addressing isolated issues, we investigate the entire range of challenges that users encounter when engaging with HF technologies whether in training, deploying, loading models, or working with HF datasets. Additionally, we delve into the complexities specific to using HF's large language models, offering a holistic perspective on the difficulties developers face throughout the various stages of model development and deployment. Moreover, previous studies have largely relied on sources such as the HF Forums, interviews, and GitHub repositories, whereas our study uniquely utilizes SO as a key source of real-world insights. This is a significant departure from existing literature, as no prior work has leveraged SO data to systematically analyze discussion topics, trends, user engagement, or the annual growth of HF-related tags. By examining SO posts from February 2019 to October 2024, we capture the most extensive range of HF-related discussions to date, offering fresh, up-to-date insights into the practical difficulties developers face when using HF technologies.

VI. THREATS TO VALIDITY

Internal Threats to Validity concern the causal relationship between treatment and outcome. Specifically, we relied on the presence of the term "*huggingface*" in the tags to identify HF-related tags on SO. Since the keyword "*huggingface*" has no direct synonyms, we believe that its use improves the precision of our data collection by ensuring that we capture only questions genuinely related to related to HF. This significantly reduces the likelihood of false positives (i.e., selecting questions that are not truly related to HF). Therefore, we believe that our method produced a highly relevant dataset. We also excluded comments, consistent with prior studies [46], since SO comments are often temporary and not available to all users. Internal threats include potential subjectivity in labeling topics after applying the LDA model. To minimize this, we followed previous works [23][53], where three authors individually reviewed keywords and SO posts, refining topic labels through several discussions to ensure representativeness.

Construct Threats to Validity concern the relation between the theory and the observations. One potential threat is the use of LDA to cluster HF questions, which could impact the construct validity. However, as noted earlier, this technique is commonly employed in similar research contexts [54][11]. Additionally, the search space used for tuning LDA parameters may introduce bias, as different parameter ranges could lead to varying results. To mitigate this risk, we used genetic search and a robust optimization library, OPTUNA, which is widely used for the automatic hyperparameter tuning of ML models [55][56]. Another potential threat involves the metrics used to assess the popularity and difficulty of the topics. However, we relied our metrics are consistent with previous studies that use similar measurements [47][23].

External Threats to Validity relate to the extent to which our findings can be generalized. While our study focuses on data collected from posts on SO, there are other forums, such as the HF Hub forum, where developers discuss HF-related topics. However, using SO enhances the generalizability of our results, as it is a highly popular platform that hosts the largest number of questions and answers from developers across diverse domains and expertise levels. We also recognize that this study could be further strengthened by incorporating data from other forums, as well as through interviews or surveys. Finally, it is important to acknowledge that our findings capture a snapshot of the current data on SO, offering a basis for future research, such as replication studies, to investigate the evolution of this field across other Q&A platforms.

VII. CONCLUSION

We present in this work an empirical study on the trends in HF-related discussions on SO and the challenges that users face. By analyzing a large dataset of SO questions, we uncovered evolving trends in HF-related discussions. Our findings revealed a significant increase in the number of HF-related questions on SO from 2019 to 2024. Also, we identified eight main topics within the discussions. In terms of popularity, our analysis revealed that HF users are more likely to post questions about *"LLMs Usage and Understanding"*. Regarding difficulty, the topic *"Distributed Computing and Resource Management"* of PTMs stands out as the most challenging topic for developers. Our findings provide a valuable resource for practitioners and researchers offering a deeper understanding of the complexities surrounding HF. We recommend that practitioners leverage community-driven documentation and troubleshooting guides to address common challenges efficiently. For researchers, we suggest investigating new methods to automate distributed computing configurations and enhance resource management for HF-related models. As future work, we aim to broaden our research to additional Q&A platforms and conduct community interviews to pinpoint HF-specific challenges and best practices in using HF technologies.

REFERENCES

[1] J. Castaño, S. Martínez-Fernández, X. Franch, and J. Bogner, "Exploring the carbon footprint of hugging face's ml models: A repository mining study," in *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 2023, pp. 1–12.

[2] W. Jiang, N. Synovic, M. Hyatt, T. R. Schorlemmer, R. Sethi, Y.-H. Lu, G. K. Thiruvathukal, and J. C. Davis, "An empirical study of pre-trained model reuse in the hugging face deep learning model registry," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, 2023, pp. 2463–2475.

[3] R. Pan, S. Biswas, M. Chakraborty, B. D. Cruz, and H. Rajan, "An empirical study on the bugs found while reusing pre-trained natural language processing models," *arXiv preprint arXiv:2212.00105*, 2022.

[4] StackExchange, "Stack exchange api documentation," https://api.stackexchange.com/docs, accessed: October 06, 2024.

[5] BeautifulSoup, "Beautifulsoup documentation," https://www.crummy.com/software/BeautifulSoup/bs4/doc, accessed: October 08, 2024.

[6] Python, "Re documentation," https://docs.python.org/3/library/re.html, accessed: October 08, 2024.

[7] NLTK, "Nltk documentation," https://www.nltk.org/, accessed: October 08, 2024.

[8] Python, "Gensim documentation," https://pypi.org/project/gensim/, accessed: October 08, 2024.

[9] Gensim, "Gensim lda," https://radimrehurek.com/gensim/models/ldamodel.html, accessed: October 26, 2024.

[10] M. Begoug, N. Bessghaier, A. Ouni, E. A. AlOmar, and M. W. Mkaouer, "What do infrastructure-as-code practitioners discuss: An empirical study on stack overflow," in *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 2023, pp. 1–12.

[11] A. Ouni, I. Saidani, E. Alomar, and M. W. Mkaouer, "An empirical study on continuous integration trends, topics and challenges in stack overflow," in *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*, 2023, pp. 141–151.

[12] H. Li, T.-H. Chen, W. Shang, and A. E. Hassan, "Studying software logging using topic models," *Empirical Software Engineering*, vol. 23, pp. 2655–2694, 2018.

[13] T.-H. Chen, S. W. Thomas, and A. E. Hassan, "A survey on the use of topic models when mining software repositories," *Empirical Software Engineering*, vol. 21, pp. 1843–1919, 2016.

[14] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[15] A. Panichella, B. Dit, R. Oliveto, M. Di Penta, D. Poshynanyk, and A. De Lucia, "How to effectively use topic models for software engineering tasks? an approach based on genetic algorithms," in *2013 35th International conference on software engineering (ICSE)*. IEEE, 2013, pp. 522–531.

[16] A. Panichella, "A systematic comparison of search-based approaches for lda hyperparameter tuning," *Information and Software Technology*, vol. 130, p. 106411, 2021.

[17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[18] H. Rahimi, J. L. Hoover, D. Mimno, H. Naacke, C. Constantin, and B. Amann, "Contextualized topic coherence metrics," *arXiv preprint arXiv:2305.14587*, 2023.

[19] M. Bhattacharya, C. Jurkovitz, and H. Shatkay, "Identifying patterns of associated-conditions through topic models of electronic medical records," in *2016 IEEE international conference on bioinformatics and biomedicine (BIBM)*. IEEE, 2016, pp. 466–469.

[20] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," in *Parallel Problem Solving from Nature-PPSN VIII: 8th International Conference, Birmingham, UK, September 18-22, 2004. Proceedings 8*. Springer, 2004, pp. 722–731.

[21] S. Messaoudi, A. Panichella, D. Bianculli, L. Briand, and R. Sasnauskas, "A search-based approach for accurate identification of log message formats," in *Proceedings of the 26th Conference on Program Comprehension*, 2018, pp. 167–177.

[22] Z. Aghababaeyan, M. Abdellatif, M. Dadkhah, and L. Briand, "Deepgd: A multi-objective black-box test selection approach for deep neural networks," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 6, pp. 1–29, 2024.

[23] A. Abdellatif, D. Costa, K. Badran, R. Abdalkareem, and E. Shihab, "Challenges in chatbot development: A study of stack overflow posts," in *Proceedings of the 17th international conference on mining software repositories*, 2020, pp. 174–185.

[24] C. Rosen and E. Shihab, "What are mobile developers asking about? a large scale study using stack overflow," *Empirical Software Engineering*, vol. 21, pp. 1192–1223, 2016.

[25] J. Castaño, M.-F. Silverio, X. Franch, and J. Bogner, "Analyzing the evolution and maintenance of ml models on hugging face," in *Proceedings of the 21st International Conference on Mining Software Repositories*, 2024, pp. 607–618.

[26] StackOverflow, "Stack overflow quote," https://stackoverflow.com/questions/65806586/valueerror-shape-mismatch-the-shape-of-labels-received-1-should-equal-the, accessed: October 25, 2024.

[27] S. Overflow, "Stack overflow quote," https://stackoverflow.com/questions/68691450/how-can-i-check-a-confusion-matrix-after-fine-tuning-with-custom-datasets, accessed: October 25, 2024.

[28] H. Face, "Hugging face spaces," https://huggingface.co/spaces, accessed: October 15, 2024.

[29] HuggingFace, "Hugging face inference endpoints," https://huggingface.co/inference-endpoints/dedicated, accessed: October 22, 2024.

[30] StackOverflow, "Stack overflow quote," https://stackoverflow.com/questions/77576705/invoking-sagemaker-endpoint-failed-since-timm-library-was-not-found-in-the-envir, accessed: October 25, 2024.

[31] S. Overflow, "Stack overflow quote," https://stackoverflow.com/questions/77595308/installation-error-with-pip-install-tokenizers-0-12-1-compatibility-issue-wit, accessed: October 25, 2024.

[32] H. Face, "Hugging face datasets library," https://huggingface.co/docs/datasets/en/index, accessed: November 04, 2024.

[33] S. Overflow, "Stack overflow quote," https://stackoverflow.com/questions/76227219/can-i-convert-an-iterabledataset-to-dataset, accessed: October 25, 2024.

[34] Huggingface, "Huggingface transformers library," https://huggingface.co/docs/transformers/en/index, accessed: October 04, 2024.

[35] S. Overflow, "Stack overflow quote," https://stackoverflow.com/questions/77044747/why-does-hugging-faces-push-to-hub-convert-saved-models-to-bin-instead-of-usin, accessed: October 25, 2024.

[36] H. Face, "Hugging face accelerate library," https://huggingface.co/docs/accelerate/en/index, accessed: October 01, 2024.

[37] HuggingFace, "Accelerate documentation," https://huggingface.co/docs/accelerate/index, accessed: October 04, 2024.

[38] S. Overflow, "Stack overflow quote," https://stackoverflow.com/questions/78865570/huggingface-accelerate-device-error-when-running-evaluation, accessed: October 25, 2024.

[39] T. B. Brown, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[40] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar *et al.*, "Universal sentence encoder for english," in *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, 2018, pp. 169–174.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[42] S. Overflow, "Stack overflow quote," https://stackoverflow.com/questions/74021562/bert-without-positional-embeddings, accessed: October 25, 2024.

[43] J. Webster and C. Kit, "Tokenization as the initial phase in nlp," in *Proceedings of the 14th conference on Computational linguistics-Volume 4*. Association for Computational Linguistics, 1992, pp. 1106–1110. [Online]. Available: https://doi.org/10.3115/992133.992154

[44] StackOverflow, "Stack overflow quote," https://stackoverflow.com/questions/77304201/replace-special-unusedx-tokens-in-a-tokenizer-to-add-domain-specific-words-in, accessed: October 25, 2024.

[45] S. Overflow, "Stack overflow quote," https://stackoverflow.com/questions/75977316/how-to-use-output-from-t5-model-to-replace-masked-tokens-in-input-sequence, accessed: October 25, 2024.

[46] P. Anthony, S. Simmons, E. A. AlOmar, C. D. Newman, M. M. Wiem, and O. Ali, "How do i refactor this? an empirical study on refactoring trends and topics in stack overflow," *Empirical Software Engineering*, vol. 27, no. 1, 2022.

[47] M. Bagherzadeh and R. Khatchadourian, "Going big: a large-scale study on what big data developers ask," in *Proceedings of the 2019 27th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering*, 2019, pp. 432–442.

[48] M. A. Batoun, M. Ennajih, M. Sayagh, and A. Ouni, "What do developers discuss about software monitoring in stack overflow? a case study on prometheus," *A Case Study on Prometheus*.

[49] A. Kathikar, A. Nair, B. Lazarine, A. Sachdeva, and S. Samtani, "Assessing the vulnerabilities of the open-source artificial intelligence (ai) landscape: A large-scale analysis of the hugging face platform," in *2023 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2023, pp. 1–6.

[50] A. Ait, J. L. C. Izquierdo, and J. Cabot, "Hfcommunity: A tool to analyze the hugging face hub community," in *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2023, pp. 728–732.

[51] A. McMillan-Major, S. Osei, J. D. Rodriguez, P. S. Ammanamanchi, S. Gehrmann, and Y. Jernite, "Reusable templates and guides for documenting datasets and models for natural language processing and generation: A case study of the huggingface and gem data and model cards," *arXiv preprint arXiv:2108.07374*, 2021.

[52] M. Taraghi, G. Dorcelus, A. Foundjem, F. Tambon, and F. Khomh, "Deep learning model reuse in the huggingface community: Challenges, benefit and trends," *arXiv preprint arXiv:2401.13177*, 2024.

[53] S. Ahmed and M. Bagherzadeh, "What do concurrency developers ask about? a large-scale study using stack overflow," in *Proceedings of the 12th ACM/IEEE international symposium on empirical software engineering and measurement*, 2018, pp. 1–10.

[54] G. L. Scoccia, P. Migliarini, and M. Autili, "Challenges in developing desktop web apps: a study of stack overflow and github," in *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*. IEEE, 2021, pp. 271–282.

[55] S. Shekhar, A. Bansode, and A. Salim, "A comparative study of hyperparameter optimization tools," in *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*. IEEE, 2021, pp. 1–6.

[56] M. Sipper, "High per parameter: A large-scale study of hyperparameter tuning for machine learning algorithms," *Algorithms*, vol. 15, no. 9, p. 315, 2022.