A Thought-provoking Question Matrix to Guide the Development of Foundation-Model-based Applications

Sietske Tacoma[1,*], Jimmy Mulder[1,†], Matthieu Laneuville[2,†] and Stefan Leijnen[1,†]

[1] *Utrecht University of Applied Sciences, Heidelberglaan 15, 3584 CS, Utrecht, The Netherlands*
[2] *SURF, Moreelsepark 48, 3511 EP, Utrecht, The Netherlands*

Abstract

Organizations feel an urgency to develop and implement applications based on foundation models: AI-models that have been trained on large-scale general data and can be finetuned to domain-specific tasks. In this process organizations face many questions, regarding model training and deployment, but also concerning added business value, implementation risks and governance. They express a need for guidance to answer these questions in a suitable and responsible way. We intend to offer such guidance by the question matrix presented in this paper. The question matrix is adjusted from the model card, to match well with development of AI-applications rather than AI-models. First pilots with the question matrix revealed that it elicited discussions among developers and helped developers explicate their choices and intentions during development.

Keywords

foundation models, use cases, model cards

1. Introduction

With the recent advent of foundation models, defined as general purpose AI-models that have been trained on large-scale data, organizations are more eager than ever to develop AI-powered applications. Foundation models have quickly built a reputation as powerful building blocks for domain-specific applications, by diminishing the need to explicate the logic needed for such applications [1]. They perform well on numerous general tasks such as text and image generation, speech recognition and graph creation [2]. Furthermore, with only limited further training, they can quickly outperform more traditional AI-models on a wide variety of domain-specific tasks. It is no wonder that organizations see the potential of foundation models and feel an urgency to explore use cases in which foundation models can be of added value for their organization.

[1*] Corresponding author.
[†] These authors contributed equally.
sietske.tacoma@hu.nl (S. Tacoma); jimmy.mulder@hu.nl (J. Mulder); matthieu.laneuville@surf.nl (M. Laneuville); stefan.leijnen@hu.nl (S. Leijnen)
0000-0002-9662-8489 (S. Tacoma); 0000-0001-9681-863X (J. Mulder); 0000-0001-6022-0046 (M. Laneuville); 0000-0002-4411-649X (S. Leijnen)

Developing applications based on foundation models also raises many questions and challenges for organizations. These include short-term questions, such as whether to use available foundation models or to train an own model from scratch, whether to use an existing model as is or to finetune it with own data, and in the latter case, which data to use. Evaluating performance is also a challenge, as the capabilities of the foundation model that have been demonstrated on benchmarks may be quite distant from the capabilities required in the organization's use case. Long-term strategic topics, such as added business value, risks and governance, are also a concern [3]. Added value can be conceptualized financially, in terms of return-on-investment, but also more generally, in terms of for example efficiency, effectivity and job satisfaction of the people using the applications. Regarding risks and governance, organizations have concerns about the dependency on models provided by Big Tech companies such as Microsoft (OpenAI), Google (Deepmind), and Amazon (Anthropic), the transparency of and possible bias in these models and the transference of intellectual property, especially when prompting or finetuning these models with own data.

Organizations are looking for guidance in addressing these questions and concerns. More specifically, once a use case has been identified and the decision has been made to start developing an application based on a foundation model, organizations are looking for ways to make responsible choices in this process. Many of these choices involve considering several options and weighing several perspectives (e.g., performance, financial and ethical aspects). In this paper we present a question matrix to guide reflection on these choices from different perspectives. By using this question matrix repeatedly during application development, developers are encouraged to explicate the options and considerations they have and to track the development of their thinking over time. This has the potential to foster 1) more deliberate choices in the designed application, both in terms of the perspectives considered as well as in both short-term and long-term benefits; 2) transparency about the design of the application; and 3) traceability which enables reuse of datasets, models, and other components in designing other, similar applications within the organization.

In this paper, we describe the design of and first experiences with this question matrix. We have used model cards [4] as a basis for the question matrix, as further elaborated in section 2. How we have transformed the model card structure into the question matrix is described in section 3. Section 4 gives an overview of our first experiences with the question matrix. In section 5 we present our conclusions and directions for further research.

2. Literature review: documentation approaches as the basis

When releasing AI-models, it is common practice to provide documentation with it, about the model's architecture, the (type of) data it was trained on and evaluated with, and its intended use. Such documentation fosters transparency of AI models and serves as a basis for assessment regarding compliance with legal requirements [5]. Documenting the characteristics of the released model asks for explicating and motivating the choices that have been made during development. Hence, such documentation approaches foster reflection on these choices. Therefore, documentation approaches could serve as a solid

starting point for designing an instrument to facilitate making these choices in a responsible way.

Most documentation approaches that have been proposed focus on data and AI models, rather than AI-systems or AI-based applications. Therefore, we chose to base our instrument on a seminal approach for documenting AI models, the model card [4]. The model card approach was proposed as a framework to report on model performance characteristics and to clarify which use cases the released machine learning model is and is not intended for. An appealing characteristic of the model card is that it asks for a description of contextual factors: the variety in groups, instrumentation, and environmental factors that the model has been evaluated on. Addressing and explicating this variety can spur reflection on inclusion and diversity during development.

The model card is an example of an information sheet: a structured collection and presentation of information on different technical and non-technical aspects. Micheli and colleagues have identified three other main categories of documentation approaches: questionnaires, composable widgets and checklists. For the purpose of guiding development, and prompting discussion and reflection, questionnaires and checklists are generally more appropriate than information sheets [6]. Especially questionnaires provide more in-depth coverage and hence encourage solid reflection about the use and potential misuse of the AI-model or system under consideration [5].

3. Development of the question matrix

As argued above, the model card structure provided a solid basis for an instrument to guide the development of AI-applications based on foundation models. This basis had to be expanded for two reasons. First, to suit AI-powered applications rather than AI models only, additional categories were needed to address the deployment and implementation of such applications. Second, to adjust the instrument for the purpose of providing guidance during development, rather than post-development documentation only, we reshaped it into a question matrix instead of an information sheet. In the next two subsections, we elaborate on these two adjustments.

3.1. Additional categories for AI-powered applications

The model card structure consists of nine categories: Model details, Intended use, Factors, Metrics, Evaluation data, Training data, Quantitative analyses, Ethical considerations, and Caveats and Recommendations. Except for model details such as model date and version, all these categories are relevant for the purpose of providing guidance during application development. Inspiration for additional categories to address deployment and implementation of AI-applications was drawn from two dominant frameworks for AI-deployment and integration: CRISP-DM [7, 8] and ML-Ops [9].

The CRISP-DM cycle consists of six phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment. While Data Preparation and Modeling were judged to be fully addressed by the model card structure, for all other phases additional items were needed. For Business Understanding, additional items concerned the use case, and more specifically the aim of developing the application, specific tasks of the

application, and the context in which the application was te be used. Furthermore, an item was added on the intended role of the application in the users' daily working processes. For Data Understanding we decided to add an item on data quality. For the Evaluation phase, we added a more general evaluation item besides the technical metrics for model performance, to evaluate whether the application indeed is appropriate for the task it was intended for. Finally, Deployment was not yet addressed in the model card, so items regarding maintenance and the embedding in the organization's software systems were added.

From the ML-Ops perspective, two additional themes where identified: future monitoring of model performance and addition of new data. Therefore, items addressing future monitoring and training new model versions were added.

3.2. Shaping the instrument into a question matrix

The model card consists of a list of items, divided into several categories. To prompt discussion and reflection, we reshaped the items into questions. Furthermore, we included multiple columns, thus shaping the instrument as a question matrix rather than as a questionnaire. The matrix consists of five categories: 1) Intended use, 2) Model properties, 3) Training, model performance and application performance, 4) Scope of the application (contextual factors), and 5) Implementation, maintenance and development. The first column resembles the model card: by answering the questions, developers give an overview of the current status of the AI-application under development. The second column asks to motivate the choices that have been made and to specify considerations that led to certain choices. The third column asks developers for the alternatives that they are considering or have considered during development.

The obtained question matrix was presented to two experts in the field of AI. They suggested that addressing internal organization, especially stakeholders that are to make decisions regarding implementation, would be useful, as these factors could also influence the choices that developers make. Adding these questions resulted in the final question matrix, of which all questions are presented in Appendix A.

4. First experiences with the question matrix

The question matrix was first piloted in three Dutch media organizations. In each of these organizations a foundation-model-based application was developed. During the development, in each organization the first author conducted a one-hour interview with an involved AI-developer, using the question matrix as interview guide. In one project, a foundation model was finetuned to be adapted for a specific purpose. The other two projects concentrated on using foundation models as offered and evaluating their performance on the organization's data for specific purposes.

In all interviews approximately half an hour was needed for specifying the intended use and the datasets needed for using or finetuning the foundation models. For these topics, the interviewees generally knew which alternatives had been considered and how choices had been made. They also were clear about their choices for the foundation models that had been selected for experimentation and development.

Concerning evaluation metrics, added value, scope, implementation, maintenance and development, their answers were less clear and complete. By analyzing the interview transcripts, three types of less concrete answers were identified. First, interviewees seemed to explicate ideas for the first time during the interview. For example, interviewees used phrases such as "Now that I think of it" and "We didn't mention it explicitly, but I think so." In multiple cases, this happened for the questions concerning what was in scope and out-of-scope for the application. Interviewees did not seem to have addressed this in their discussions with colleagues, but did appear to have implicit ideas about what was beyond the scope of their application, which they explicated during the interviews. Second, interviewees identified topics that had not been addressed yet in development and needed attention. This was especially the case for more technical topics such as the use of specific evaluation metrics and the way in which cross-validation could or should be used in the finetuning procedure. An interviewee pondered that "maybe these are questions that we should take into the organization", expressing a realization that more attention for these topics was needed and fellow developers and other stakeholders within the organization should be involved. Third, interviewees started developing new ideas during the interview. This especially happened in an interview with two interviewees, were answers by one interviewee seemed to ignite new ideas in the other. This shows that using the question matrix in development teams may help teams to explicate ideas, develop a shared understanding of these ideas and build on each others ideas.

5. Conclusions and future research directions

In this paper we presented a question matrix that is aimed at helping developers explicate their options and the consequences of their choices repeatedly during development of foundation-model-based applications. The question matrix is based on seminal approaches for documenting AI-models, and adjusted to apply to AI-applications by drawing from literature on CRISP-DM and ML-Ops. First experiences with the question matrix show that it indeed seems to encourage discussion and reflection during development. To exploit this potential, we envision that developers fill in the question matrix repeatedly during the development and deployment of a foundation-model-based application, for example at the beginning and halfway through the development project, towards the deployment phase and repeatedly during deployment.

We conjecture that filling in the question matrix also serves well as a documentation approach, especially within organizations. It fosters transparency of these applications and could enable easier reuse of data, (foundation) models and architectures for other purposes within the organization. Further research is needed to address this potential.

Another direction for future research is the completeness of this question matrix. Organizations express a desire that an instrument like this may help them avert or mitigate future risks, such as dependence on Big Tech companies and bias caused by foundation models. Using, for instance, separate ethics checklists may feel like an extra burden. Therefore, in the question matrix we have aimed to address AI-application development from multiple perspectives and throughout its lifecycle, to obtain a sense of completeness. Future research is needed to further develop and assess this completeness, for example by

aligning the instrument with the practice of regulatory oversight, as will be required by the AI Act. As regulatory oversight may differ between sectors, this may lead to tailored question matrices for different sectors. Hence, evaluation of the question matrix and its completeness in various sectors is also a promising venue towards more responsible implementation of foundation-model-based applications.

References

[1] R. Bommasani *et al.*, "On the Opportunities and Risks of Foundation Models," Aug. 2021, doi: 10.48550/arxiv.2108.07258.

[2] C. Zhou *et al.*, "A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT," Feb. 2023, Accessed: Mar. 21, 2024. [Online]. Available: http://arxiv.org/abs/2302.09419

[3] S. Leijnen, H. Aldewereld, R. van Belkom, R. Bijvank, and R. Ossewaarde, "An agile framework for trustworthy AI.," *NeHuAI@ ECAI*, pp. 75–78, 2020, Accessed: Apr. 15, 2024. [Online]. Available: https://www.academia.edu/download/76467298/leijnen.pdf

[4] M. Mitchell *et al.*, "Model cards for model reporting," *FAT* 2019 - Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency*, pp. 220–229, Jan. 2019, doi: 10.1145/3287560.3287596.

[5] M. Micheli, I. Hupont, B. Delipetrev, and J. Soler-Garrido, "The landscape of data and AI documentation approaches in the European policy context," *Ethics Inf Technol*, vol. 25, no. 4, Dec. 2023, doi: 10.1007/S10676-023-09725-7.

[6] M. A. Madaio, L. Stark, J. Wortman Vaughan, and H. Wallach, "Co-Designing Checklists to Understand Organizational Challenges and Opportunities around Fairness in AI," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, Apr. 2020, pp. 1–14. doi: 10.1145/3313831.3376445.

[7] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying CRISP-DM process model," *Procedia Comput Sci*, vol. 181, pp. 526–534, 2021, doi: 10.1016/j.procs.2021.01.199.

[8] P. Chapman, "CRISP-DM 1.0 Step-by-step data mining guide," 2000, Accessed: Mar. 22, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:59777418

[9] D. Kreuzberger, N. Kuhl, and S. Hirschl, "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," *IEEE Access*, vol. 11, pp. 31866–31879, 2023, doi: 10.1109/ACCESS.2023.3262138.

A. The questions in the question matrix

Questions in the Question matrix
Intended use Questions in this category concern the intended use of the AI-application under development.
Purpose
With what purpose is the application being developed?
What is the task the application is supposed to carry out?
In which context or situation is the application supposed to be used?
Which tasks are out-of-scope? So what is the application not intended for, while users might think it is?
Intended users
Who are the intended users of the application?
What is the size of the user group?
In what way are the intended users involved in the development process?
Integration into working routines
How are the users supposed to integrate the application into their working routines?
How often are the users supposed to use the application?
How does the application relate to other applications that the user uses?
Model properties Questions in this category concern the AI-model that is employed within the AI-application under development.
Architecture
What type of model architecture is being used in the application?
Which foundation model(s) is/are used within the application?
Are you using the foundation model as is, do you finetune the model using training data, or do you adjust the architecture of the foundation model?
Are there hyperparameters to tune and, if so, which values are chosen?
Training data - only needed to answer when finetuning or adjusting an existing model, not when using an existing model as is within the application
Which dataset(s) is/are being used for training or finetuning the model?
How are the training dataset and its annotation created?
How is the training data preprocessed?
What is the quality of the training data?

Which selection criteria for including data in the training data set are used?
Developers
Who is developing the AI-application?
Which parts of the application are developed within your organization and which parts are developed by other parties?
Training, model performance and application performance
Metrics
Which metrics does your organization use to evaluate the model?
To what extent do you identify and monitor metrics for various groups or categories (also see Scope of the application)?
If applicable: which decision tresholds are being used?
Which amount of variation is present in the values of the evaluation metrics?
How do you evaluate whether the application indeed is appropriate for the tasks you have identified for it?
Training procedure - only needed to answer when finetuning or adjusting an existing model, not when using an existing model as is within the application
What does the training procedure look like?
(In which way) is cross-validation used?
Do you combine results of multiple runs?
Evaluation data
Which dataset(s) is/are used to evaluate the model? application?
How are the evaluation dataset (and its annotation) created?
How is the evaluation data preprocessed?
How do you make sure the evaluation dataset is appropriate for evaluation (taking into account contextual factors and representativity)?
Scope of the application (contextual factors)
Groups
For which different groups (e.g. cultural, demographic, phenotypic) should the application perform?
How are these different groups taken into account in training data, training procedure and evaluation?
Instrumentation
For which variation in instrumentation (e.g. image quality, sound quality) should the application perform?
How are these different instrumentations taken into account in training data, training procedure and evaluation?
Environment

	For which variation in environmental factors (e.g. light, weather conditions) should the application perform?
	How are these different environmental factors taken into account in training data, training procedure and evaluation?
Implementation, maintenance and development	
	Implementation
	How is the application supposed to be implemented within the organization?
	Who decides on actual implementation?
	Maintenance
	How is the application supposed to be maintained?
	Who are involved in maintaining the application?
	How will be monitored whether the model keeps performing as intended and whether model drift or model shift occurs?
	What is your plan for identifying and mitigating risks?
	Development
	How and how often is a new version of the model trained?
	How do you handle newly available data?
	Who are involved in further development of the application?