34th Annual INCOSE international symposium
hybrid event
Dublin, Ireland
July 2 - 6, 2024

Integrating AI with MBSE for Data Extraction from Medical Standards

Ibrahim Ghanawi
SysDICE GmbH
Franz-Volhard-Str. 5
68167 Mannheim, Germany
ibrahim.ghanawi@sysdice.com

Mohammad Wissam Chami
SysDICE GmbH
Franz-Volhard-Str. 5
68167 Mannheim, Germany
mohammad.w.chami@sysdice.com

Mohammad Chami
SysDICE GmbH
Franz-Volhard-Str. 5
68167 Mannheim, Germany
mohammad.chami@sysdice.com

Marko Coric
Mechatronic GmbH
Europaplatz 5
64293 Darmstadt, Germany
marko.coric@mechatronic.de

Nabil Abdoun
SysDICE GmbH
Franz-Volhard-Str. 5
68167 Mannheim, Germany
nabil.abdoun@sysdice.com

Abstract. The growing adoption of Model-Based Systems Engineering (MBSE) in the medical sector has prompted a significant emphasis on the digitization of medical standards into norm models. This transformation promotes consistency and allows for tracing system model elements to the corresponding norm model elements. Despite these efforts, the current digitization activities heavily rely on manual extraction and transformation, particularly from PDF documents into SysML models. Concurrently, the proliferation of Artificial Intelligence (AI) applications in recent years presents an opportunity to automate such activities. This paper contributes to the integration of AI with MBSE, focusing solely on the extraction and transformation of medical standards information from documents into SysML norm models. It explores the initial outcomes of augmenting data extraction from medical standards using recent AI algorithms and integrating them into MBSE practices. The evaluation involves two approaches, an open-source multimodal classifier model and a proprietary large language model. The study assesses these approaches on a medical standard and outlines future work, including the exploration of an open-source large language model approach.

Keywords. Model-Based Systems Engineering, Artificial Intelligence, Digitization, Norm Compliance, Large Language Model, Classification.

Background and Problem Description

The domain of Systems Engineering (SE) (Sillitto et al., 2019) is practiced in industry to deal with an interdisciplinary process for supporting the entire system life cycle. A significant evolution within SE is the emergence of Model-Based Systems Engineering (MBSE), which represents the formalized application of modeling to support diverse SE activities (INCOSE, 2021). MBSE encompasses multiple modeling concepts: modeling language, modeling method, and modeling tool, employed to create one or more comprehensive systems models. A systems model encapsulates crucial model elements (e.g., requirements, functions, test cases…) along with their relationships (e.g., satisfy, refine, allocate...).

Despite the notable success stories about MBSE, its adoption in real-world applications faces substantial challenges (Karban et al., 2011) (Chami & Bruel, 2018) where neither the MB nor the SE aspect can handle it. In the sequel of our AI4MBSE work, we presented in (Chami et al., 2022) the vision capabilities of an AI4MBSE solution for solving a set of MBSE challenges and demonstrated in (Abdoun & Chami, 2022) the application of natural language processing (NLP) techniques for automatic text classification of PDF documents.

The work presented in this paper is part of the CyberTech research project (CyberTech, 2024), which focusses on the advanced systems engineering for the design of cyber-physical systems. The overall aim of the research project is to develop software-based tools for data integration across the life cycle of cyber-physical systems. The project places a strong emphasis on three pillars: **human**-centered work design, socio-**technical** methods, and **organization** integrated approach. The scope of this paper dives into the integration of AI and MBSE to automate the data extraction from medical standards, ensuring their consideration throughout the development process and providing evidence of compliance post-development of medical systems.

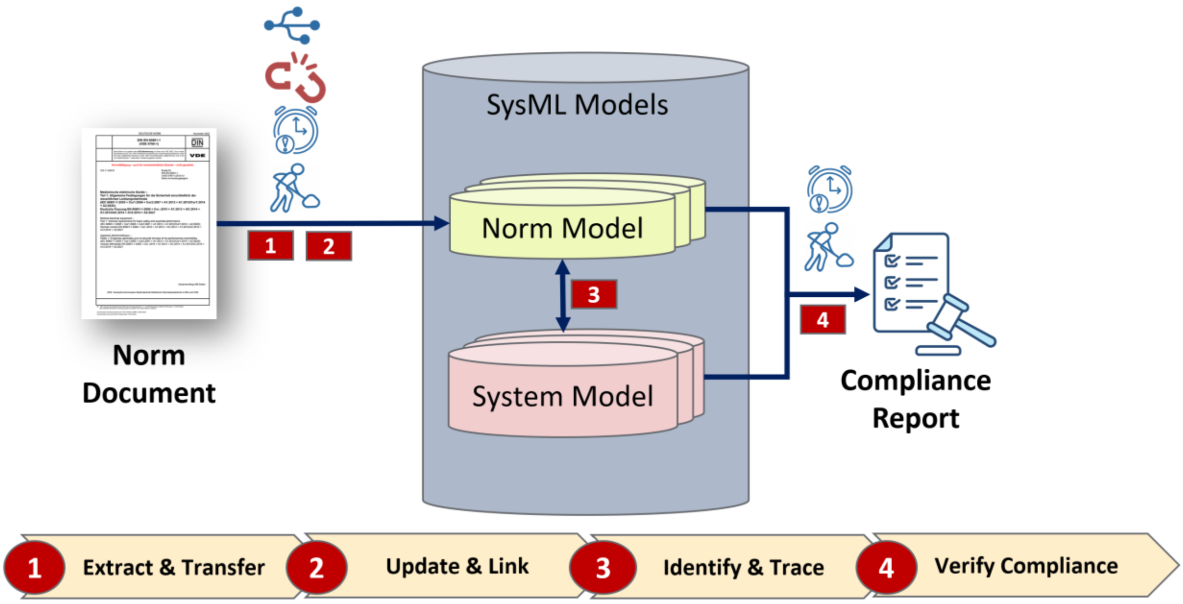

Figure 1 – The Four Use Cases of the Assisted Norm Compliance

Figure 1 illustrates the high-level project use cases of the Assisted Norm Compliance (ANC) solution. In brief, the classical approach of the norm compliance involves several activities: a manually extracting norm requirements or needs from medical standards documents into the norm model in a SysML modeling tool, manually linking norm model elements with the corresponding system model elements, and finally conducting other risk management and compliance-related tasks, e. g. verification of the norm conformity by testing, to generate the compliance report for the developed system. These activities involve collaboration among multiple stakeholders. Typically, quality management stakeholders focus on the norm documents, while system engineers and architects concentrate on the SysML models. As a result, any proposed solution should seamlessly address the integration of both documents and SysML models.

In the classical approach to norm compliance, several challenges hinder the efficiency and effectiveness of the system life cycle:

- Time-consuming and costly work associated with the **digitization of norm documents** into the norm models: The process of transforming norm requirements from document-based sources into a digital model-based content proves resource-intensive and economically burdensome. For standards such as IEC 60601-1, weeks are required for the manual extracting and transfer of relevant information to the norm model.
- Lack of **traceability between norm documents and norm models**: Without a robust mechanism of linking the source and evolution of norm requirements, maintaining a clear and auditable link becomes difficult. This linking is particularly important when different medical systems of one organization have been approved on the basis of different editions of the same standard.
- Manual **versioning and configuration management** for updates to norm documents: After any change in the content or status of the norm, e.g. new amendment or corrigenda, the whole extraction of the medical standards must be reviewed and adapted according to the change. The manual handling of versioning and configuration management for updates to norm documents adds another layer of complexity. This manual process not only consumes time but also introduces the potential for errors in managing and tracking changes.
- Time-consuming **norm compliance** and challenges in ensuring **completeness** and **gap analysis**: The process of norm compliance is not only time-consuming but also presents difficulties in ensuring the completeness of requirements and conducting gap analysis. Moreover, conducting a thorough gap analysis to identify discrepancies between norm models and system models becomes a complex and resource-intensive endeavor. For the approval of a medical system, the proof must be provided that the norms have been considered in their entirety. Secondly, the proof must be provided that the medical system also complies with the standards, e.g., by verification.
- Another challenge is the way in which medical standards are written. Standards authors, at least in the field of medical systems, do not see themselves as stakeholders of a medical device. Standards are usually written as prose texts and their semantic structure is not a collection of requirements or needs. Based on our experience, it's evident that medical standards exhibit a lesser inclination towards systems engineering principles compared to standards in domains like railway or space.

Given these challenges, a pivotal question emerges: *How can we automate procedures for norm compliance in the medical sector*? Tackling these challenges demands innovative solutions that harness advanced technologies. Although MBSE forms an essential part of the research project's solution space, the objective is to integrate the appropriate AI solutions for the gaps that MBSE is not yet able to solve.

This paper focuses on the first use case of the ANC approach shown in Figure 1: Extract and Transfer. The primary objective of the Extract and Transfer use case is to automate the manual extraction of norm requirements using AI while maintaining traceability between the data source in the norm documents and their representation on the norm models.

The rest of the paper is organized as follows: the "*Related Work"* section summarizes other related research work dealing with the data extraction topic and highlights the applied AI algorithms. The "*Evaluated Approaches"* section describes the two proposed approaches and is followed up by "*Evaluation Scheme"* and "*Evaluation Results"* that illustrate the results of adopting the proposed approaches on a medical standard. Finally, the "*Future Work"* and "*Conclusion"* sections describe a third approach and summarize the paper results.

Related Work

The field of data extraction from documents, and natural language processing have seen a huge boost in recent years. In this paragraph, we go into some related publications. As our paper extends across multiple fields, we aim to provide a concise overview of related works in each field rather than an exhaustive survey.

When it comes to data extraction from documents, (Bast & Korzen, 2017) released a benchmark highlighting available tools for text extraction from PDF Documents, the approaches we talk about in this paper use such tool to transform a document into text, before feeding this text into an AI model. Moreover, (Denk & Reisswig, 2019) proposed BertGrid, a way to represent a document as a grid of embedding vectors, containing both text and spatial information. Their proposed tool was then used for extracting data from invoices. (Yaojie et al., 2022) presents a generalized framework for information extraction (IE) tasks, known as UIE (Unified Information Extraction). Another framework was introduced by (Wang et al., 2022) named LiLT works by jointly encoding text and image data then use it to achieve downstream tasks like Entity Recognition, and Relation Extraction. (Wu et al., 2021) introduces a novel approach to information extraction (IE) called text-to-table. This method transforms text into table format, summarizing the main content of the text in a structured manner. In summary, the topic of data extraction from documents has a lot of significant contributions. Our focus revolves around enhancing the generality and simplicity of the approach, making it more industrial-ready and easier to train.

Another related field is image processing, where several papers were published for large pre-trained neural networks for multi-class classification. We will focus on "Very Deep Convolutional Networks for Large-Scale Image Recognition" (Simonyan & Zisserman, 2014), where the impact of network depth in convolutional networks on image recognition accuracy was studied. The VGG16 model released in this paper was used across the literature to assist with image processing tasks (Tammina, 2019). In our approach, we also use the VGG16 model to obtain image embeddings.

In Natural Language Processing (NLP), the transformer architecture (Vaswani et al., 2017) began a wave of capable Large Language Models (LLMs) that utilized this architecture to achieve text-to-text tasks. One example is BERT (Devlin et al., 2019), which marked a significant advancement in NLP due to a design that allows the pre-trained BERT model to be fine-tuned to achieve a wide range of tasks. After BERT, T5 (Text-to-Text Transformers) was developed by google (Raffel et al., 2019) as well as the GPT (General Pretrained Transformer) models by OpenAI, like GPT-3 (Brown et al., 2020). These models showed an ability to achieve a huge number of downstream tasks by changing the instructions in the model's text input. Since then, numerous models were released under either a proprietary (Chowdhery et al., 2022) (Bubeck et al., 2023) or an open-source license (Touvron et al., 2023) (Hoffmann et al., 2022). In our paper, we reuse the encoder part of BERT and two GPT models for our two approaches below.

Significant work has been done to make it easier to customize the behavior of pretrained LLMs to achieve better performance on a given downstream task. We here focus on LoRA (Low Rank Adaptation (Hu et al., 2021)), which is a way to reduce the resources required for training by training a lower rank matrix, that is multiplied with the weights of the select LLM layers to improve the behavior. This approach allowed for efficient training, and showed comparable results to full finetuning, without compromising the inference time, and reducing the required storage and loading time of different models. Recently, QLoRA was released (Dettmers et al., 2023), this further reduced the required resources by allowing to quantize the LLM to 4-bit during training, with a similar performance to the original LoRA. This work is relevant for our *Future Work: Open-source Large Model* section.

Finally, in the field of Systems Engineering (SE), several papers were released for processing SE information, with the majority focusing on requirements. For example, the authors in (Sharma et. Al, 2021) take a group of requirements and categorizes them into categories according to criticality. Another example is (Abualhaija et al., 2020), where a feature-based machine learning approach was proposed to extract requirements from specification documents. The existing research shows a gap between the recent AI advances and the AI applications in MBSE.

Evaluated Approaches

In this section, we give an overview of the two approaches we evaluated:

1. Open-source Multimodal Classifier Model: We take a document, classify each element, and then reconstruct the desired output from the classification results. We use both visual and text inputs for this approach.

2. Proprietary Large Language Model: We send prompts to a proprietary LLM to obtain the desired output as a text. The prompts contain text from the document along with instructions for the model on how to extract and format the needed data.

Below is a breakdown of the technical details of each approach.

Approach One for Extraction: Open-source Multimodal Classifier Model

One way to achieve the extraction is to classify each element in the document, and then extract the desired elements based on the classification results. A document contains both visual (e.g., font size, font weight, shapes) and textual information, so in this solution, both are used to classify a document's elements (multi-modal). This approach was inspired by the idea of image captioning, where image information is encoded and passed to a textual model to generate captions (Huang, 2019) (He, 2020).

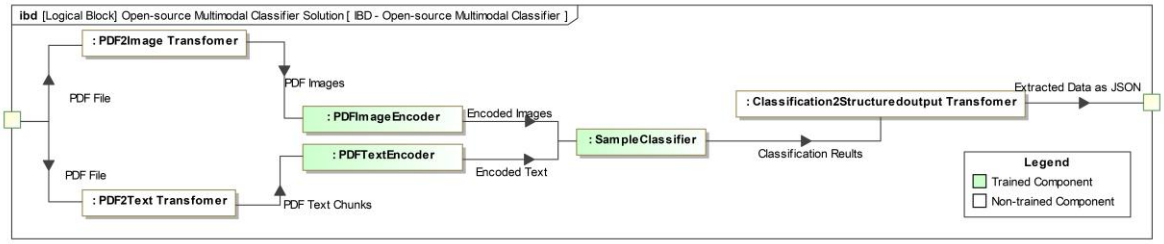

Figure 2 - Illustration of the Opens-source Multimodal classifier Solution Components

Figure 2 illustrates the components and their interactions when inferencing with such an approach. The document's pre-processing involves transforming it into images for every page, and text chunked into smaller pieces that are passed to be classified. The images and text chunks are then passed to the corresponding encoders that produce encodings for both the visual and textual information. A classifier then takes in the encodings corresponding to each text chunk. Finally, the classification output is used to generate a structured JSON object that represents the data needed for extraction. In approach one, both encoders and the classifier are trained on domain specific annotated documents.

In this paper, we used a BERT (Devlin et al., 2019) model as the text encoder; initialized from the pretrained weights, and further trained on domain specific documents. A VGG16 (Simonyan & Zisserman, 2014) model was used for the image encoder, which we also used from pre-trained and then further trained. The classifier was a multiheaded self-attention transformer (Vaswani et al., 2017) with a SoftMax activation. The implementation was done in Pytorch-lightning (Pytorch Team, 2019) for the trained components and PyMuPDF (McKie, 2015) for processing the documents.

Approach Two for Extraction: Proprietary Large Language Model

With the rise of generative AI and its influence in the last few years, the public obtained access to increasingly powerful proprietary Large Language Model, trained, fine-tuned, and aligned to perform a variety of downstream tasks (e.g., GPT3, PalM2…). Such models can be used out-of-the-box to perform data extraction from natural text, without the requirement for further training.

Although no training is performed, this approach still requires processing steps to transform the documents into text input for the LLM; and to transform the text output of the LLM into the desired JSON format. To achieve the extraction from the document, we are passing the entire document into the model, divided into pieces to account for the max tokens allowed for the used model. The components of this solution and their interaction during inference can be seen in Figure 3.

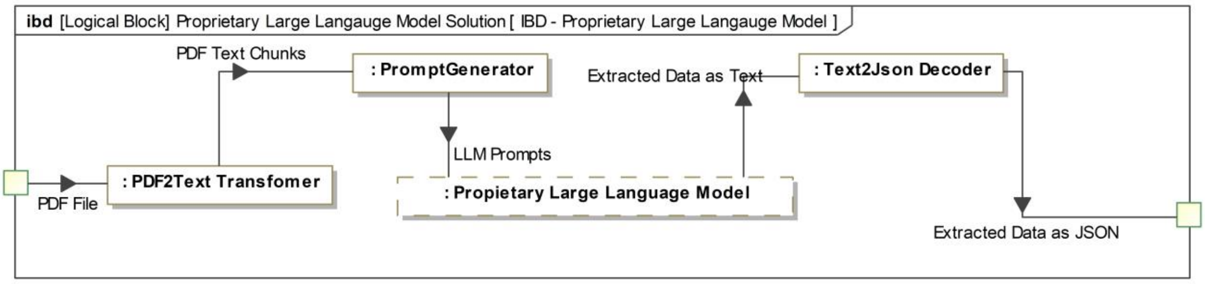

Figure 3: Illustration of the Proprietary Large Language Model Solution Components

In this paper, we evaluated two models offered by OpenAI through their public API, the "gpt-4-1106-preview" and "gpt-3.5-turbo-1106". We used few-shot prompts, prepared with LangChain (Chase, 2022). The used prompt also specifies the JSON format, so the Text2JSON Decoder only makes sure the format is valid and decodes the string into a JSON object. The documents were processed with PyPDF2 (*PyPDF2*, 2007) because it preserves the document layout better when converting the document to text (compared to PyMuPDF).

Transformation to a SysML Norm Model

In this section, we will cover how we transfer the extracted data to a SysML model. After obtaining a JSON formatted representation for the requirements from each solution, it is sent to a custom plugin in the systems modelling tool Cameo Systems Modeler (Dassault Systèmes, 2024). The plugin is responsible for processing the JSON input and generating the norm model, the mapping into SysML elements (stereotypes and properties) is predefined via static rules and aligned with the adopted modeling method.

An example of a generated norm model can be seen in Figure 4, the titles are used as the requirement name, while the content are saved in the requirement text. For ease of use, the hierarchy in the original document is preserved, and a link is added for each requirement to the location in the document to achieve the linking shown in the second use case Update and Link of Figure 1.

Figure 4 – A Snapshot for the SysML Norm Model for "BS EN 61010-12010"

The classification results from approach one are used to annotate the input PDF Document, which can be validated by a human. This process is optional but ensures that any errors during the classification phase are not propagated to the created SysML Model. The annotation is challenging to perform with the second approach, because the trace to the text position in the PDF file is lost after passing it to the proprietary LLM. We tried to solve the issue in post-processing by matching to text in the PDF, but the algorithm failed to annotate most of the elements.

Evaluation Scheme

Since we are evaluating two different approaches (classification vs. generative), it's not straight forward to derive unified metrics for evaluation. So, all the approaches were processed to have the same form in the output, which we used to derive the precision and recall metrics. The accuracy metrics were evaluated in two folds:

- First, we evaluate how well the model extracted the titles.
- Second, for the titles an approach was able to detect, we evaluate how well their contents were retrieved using a text similarity measure.

We used fuzzy string matching when calculating the precision and recall for the titles. This was done because a Generative Model may introduce some noise into the extraction, and we wanted to tolerate that on a few characters level. For the requirement content, we used BERT similarity score, we averaged the scores for the content of all the retrieved titles. The implementation for the similarity score was obtained from BERTSimilarity (Majumder, 2020).

In addition to the quality metrics, we also used other performance and cost metrics to evaluate the models. The measures used are:

- Inference time: the average time it takes the approach to extract data from a pdf document (measured in seconds per document page, sec/page)
- Inference cost: the average computing cost to extract data from a document using a given approach (measured in $/page)

Finally, since the two approaches in this paper are oriented towards industry consumption, we are compelled to consider non-technical metrics as well. These metrics and their explanation are described in the *Evaluation Results* section.

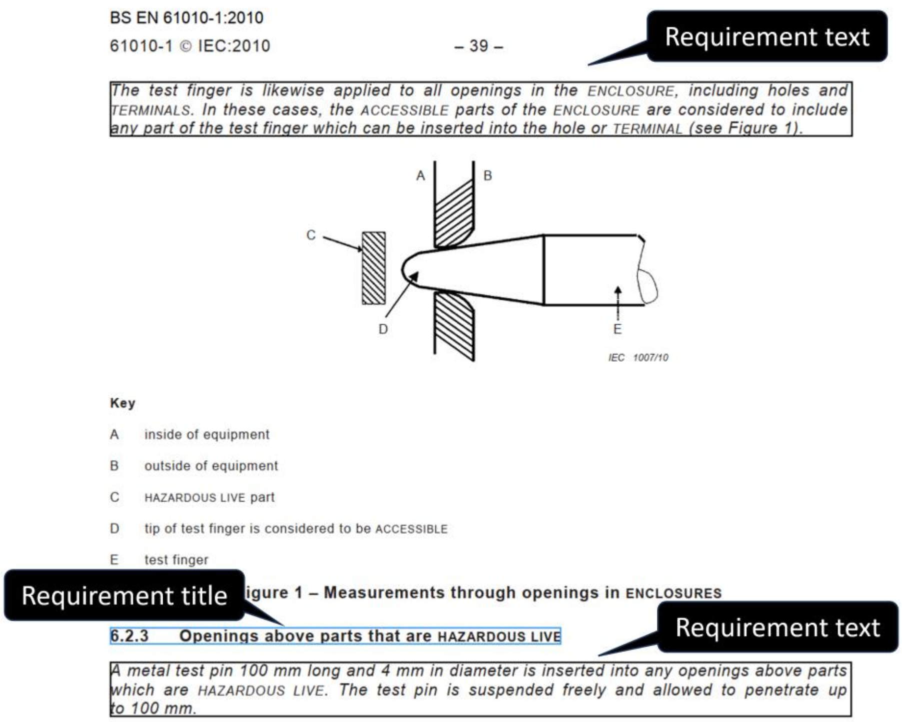

Figure 5 - Example Annotations in “BS EN 61010-12010”

For the evaluation data, we used an annotated version of the “BS EN 61010-1:2010” standard titled as “*Safety requirements for electrical equipment for measurement, control, and laboratory use*”. Nine other ETSI standards were used for training of the first approach. In each document, we had two kinds of annotations, we either annotate a title for the requirement to be extracted, or we annotate its content. We need our models to ignore all non-annotated text when extracting. Figure 5 shows an annotation example of the requirements’ title and requirements’ text.

Evaluation Results

Table 1 shows the evaluation accuracy metrics for our implementations. We see that all three implementations performed well in retrieving the existing titles (recall), but with much lower precision (more false positives). Interestingly, GPT-4 performed slightly worse than GPT-3.5 on the document used for testing, although the used prompt template was tuned on GPT-4. We also calculated an aggregated score, which is the average F-score over both fields (title and content). The classifier approach still performed much better than the LLMs. Since the LLMs only received text inputs, the models struggled with distinguishing elements that require format information, additionally, the specificity of the expected results was challenging to convey through a prompt, even when examples were added, The models identified titles correctly, but they also identified a significant number of “non-titles” as titles. Fine-tuning a proprietary LLM would improve the results, but it would also add additional costs for both training and subsequent usages.

Table 1: The Accuracy Metrics for the Compared Approaches

Implementation	Title		Content		Aggregated Score
	Recall	Precision	Recall	Precision	
BERT + VGG16 (Approach One)	100	84.57	95.92	96.06	93.81
GPT-4-1106-preview (Approach Two)	98.71	45.19	92.63	91.91	77.12
GPT-3.5-Turbo-1106 (Approach Two)	99.24	52	90.19	91.37	79.48

We also show the cost and time of inference for each approach in Table 2. As expected, the inference cost for approach one was the lowest, followed by GPT-3.5, then GPT-4. The high cost of using GPT-4 combined with its performance makes it inefficient to use for our task. The inference time followed a similar pattern as well, but the numbers here are much closer. For the Proprietary LLM approach, we sent requests asynchronously to reduce the response time per request.

Table 2: The Cost Metrics for the Compared Approaches

Model	Inference time (sec/page)	Inference cost ($/page)
BERT + VGG16 (Approach 1)	0.8	0.0001
GPT-4-1106-preview (Approach 2)	1.42	0.13
GPT-3.5-Turbo-1106 (Approach 2)	1.19	0.003

Some interesting ideas to reducing costs for LLM usages were suggested in the literature, for example FrugalGPT (Chen et al., 2023) talked about steps to significantly reduce the inference costs, however we think the ideas presented in FrugalGPT will not result in the same cost savings in our case, for example LLM approximation would be good in a Question Answering task, but for extracting data from a document, pages are not likely to be repeated. It would be interesting however to try out LLM cascades, decomposing our task into several smaller tasks handed over to various models, but we leave this out of scope of this paper.

To obtain the cost metrics for approach one, we used a single V100 GPU, the cost will depend on the hardware setup used. It's also worth noting that there is a training cost associated with the Multi-modal classifier, this involved annotating 9 documents, and 13 minutes of training on the V100 GPU. But with the repeated usage of the trained model, the divided cost will not be significant for inference.

Besides the evaluation results shown, other aspects need to be considered when addressing industrial applications. The most relevant for our research project are listed below:

- Flexibility: how easily the proposed solution can be adapted for different use cases or new documents, e.g., railway or space standards. In this metric, the proprietary approach would perform much better, as the annotation of new documents, and the limitations of the input and output shape limits the flexibility of the first approach.
- Privacy: Approach two involve sharing documents content with a third party, while the classifier can be hosted in a company's internal infrastructure, offering privacy.
- Model Upkeep and Maintenance: with a proprietary LLM, such challenges are handled by the service providers, making it more convenient to pick up and build on top.

Future Work: Open-source Large Language Model

As a future work, we plan on implementing a third approach that uses an open-source generative LLM for the extraction. With this approach, the LLM is less powerful than a Proprietary LLM, but fine-tuning it on domain specific documents would achieve better performance for our specific task.

The interaction of the solution components during inference can be seen in Figure 6. It shares most of the components with the Proprietary LLM (Figure 3). The only differences here being the open-source LLM, and a trained LoRA adaptor that is used to modify its behavior.

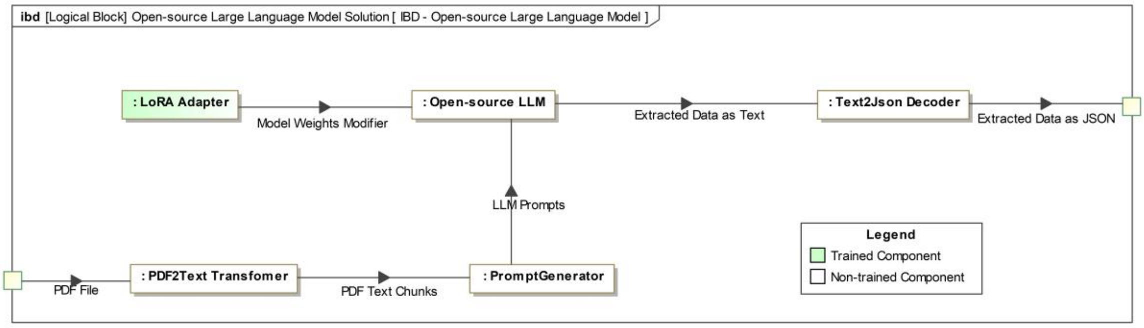

Figure 6: Illustration of the Proprietary Large Language Model Approach Components

We aim for an implementation that can be deployed on a small/medium company scale machine, with even the possibility of deploying it on a closed (no internet) limited system. So, we limited the number of parameters of the LLM to 13b parameters. Hereby, QLORA (Dettmers et al., 2023) with 4-bit quantization is the most appropriate for minimizing the resources required for finetuning. However, implementing this solution is challenging because of the following factors:

- The training samples are limited (a little above a 1000 samples)
- In most cases, the context of the model is limited to 4096 tokens, and hovering very close to this limit degrades the model performance.
- We are trying to force the model to output a specific format, which can be used to retrieve the desired output.

To overcome these challenges, pre-training on a large dataset for a similar task seems necessary. One example is PubLayNet (Zhong et al., 2019), with thousands of documents that can be used to train on extraction style tasks. This approach will be more costly compared to the multi-modal classifier approach, but it could come with the benefit of increased flexibility while maintaining a high performance.

Conclusion

In our paper, an approach to assist in the digitization of norm documents into SysML norm models is presented. We evaluated two approaches for extracting information from unstructured PDF documents in the medical sector and demonstrated the transformation of the extracted information into SysML norm models. Our findings indicate that using a classification approach can generate better results after training on a small set of annotated documents compared to out-of-the-box proprietary LLMs.

Our achievement in substantially **reducing the time required for digitizing norm documents**, while maintaining an acceptable error rate, is particularly noteworthy. This accomplishment facilitates the integration of norm compliance into the system development life cycle. Furthermore, our progress in establishing a solution for **linking norm documents to norm models** represents a significant advancement in addressing the challenge of traceability within the system development life cycle. By bridging the gap between the source and evolution of norm requirements, we have enhanced the clarity and auditability of links, which is especially crucial in scenarios involving different editions of the same standard. This achievement lays a solid foundation for ensuring consistency and accuracy in norm compliance across diverse medical systems within an organization.

For our future work, we will consider the three other project use cases within the context of "Assisted Norm Compliance", as part of the CyberTech project. This will involve matching SysML model design and architecture elements and verifying their compliance with extracted requirement. Additionally, we plan to focus on improving existing extraction approaches, from reducing the resources required for inference to enhancing performance without increasing the necessary training data.

Acknowledgments. This research and development project CyberTech is funded by the German Federal Ministry of Education and Research (BMBF) within the "Innovations for Tomorrow's Production, Services, and Work" Program (02J19B012) and implemented by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the content of this publication.

References

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, … Noah Fiedel. (2022). PaLM: Scaling Language Modeling with Pathways. *Journal of Machine Learning Research*.

Abdoun, N., & Chami, M. (2022). Automatic Text Classification of PDF Documents using NLP Techniques. 32nd Annual INCOSE International Symposium 25–30 June 2022 — Detroit, MI. https://doi.org/10.1002/iis2.12997

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin (2017). Attention is All you Need. *NIPS*.

Bast, H., & Korzen, C. (2017). A benchmark and evaluation for text extraction from PDF. *ACM/IEEE Joint Conference on Digital Libraries*, 99–108. https://doi.org/10.1109/jcdl.2017.7991564

Brown, T. B., Mann, B., Ryder, N., Nick Ryder, Subbiah, M., Kaplan, J., … Amodei, D. (2020). Language Models are Few-Shot Learners. *Neural Information Processing Systems*.

Chami, M., & Bruel, J.-M. (2018). A survey on MBSE adoption challenges. The Systems Engineering Conference of the Europe, Middle-East and Africa (EMEA) Sector of INCOSE (EMEASEC 2018). Berlin, Germany: Wiley Interscience Publications.

Chami, M., Abdoun, N., & Bruel, J.-M. (2022). Artificial Intelligence Capabilities for Effective Model-Based Systems Engineering: A Vision Paper. 32nd Annual INCOSE International Symposium 25–30 June 2022 — Detroit, MI https://doi.org/10.1002/iis2.12988

Chase, H. (2022). LangChain. Retrieved from https://github.com/langchain-ai/langchain

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv: Learning*.

CyberTech (2024). Advanced Systems Engineering for the Design of Cyber-Physical Systems. Research Project funded by the German Federal Ministry of Education and Research (BMBF). https://www.ase-cybertech.de/

Dassault Systèmes (2024), Cameo Systems Modeler Tool. Available at: https://www.3ds.com/proucts/catia/no-magic/cameo-systems-modeler

Denk, T. I., & Reisswig, C. (2019). BERTgrid: Contextualized Embedding for 2D Document Representation and Understanding. *arXiv: Computation and Language*.

Falcon, W. (2019). *Pytorch-lightning*. Pytorch Team. Retrieved from https://www.pytorchlightning.ai

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, … Thomas Scialom. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv.Org*.

INCOSE, Systems Engineering Vision 2035: Engineering Solutions for a Better World, International-Council on Systems Engineering, 2021. https://www.incose.org/publications/se-vision-2035

J. E. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, & Weizhu Chen. (2021). LoRA: Low-Rank Adaptation of Large Language Models. *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, & Kristina Toutanova. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *North American Chapter of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/n19-1423

Jiapeng Wang, Lianwen Jin, & Kai Ding. (2022). LiLT: A Simple yet Effective Language-Independent Layout Transformer for Structured Document Understanding. *Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2022.acl-long.534

Jordan Hoffmann, Sebastian Borgeaud, A. Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, … L. Sifre. (2022). Training Compute-Optimal Large Language Models. *arXiv.Org*. https://doi.org/10.48550/arxiv.2203.15556

K. Simonyan & Andrew Zisserman. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*.
Karban, R. (2011). Cookbook for MBSE with SysML. MBSE Initiative-SE2 Challenge Team.
Lingjiao Chen, M. Zaharia, & James Y. Zou. (2023). FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *arXiv.Org*. https://doi.org/10.48550/arxiv.2305.05176
Sallam Abualhaija, Chetan Arora, Mehrdad Sabetzadeh, Lionel C. Briand & Michael Traynor (2020). Automated demarcation of requirements in textual specifications: A machine learning-based approach. Empirical Software Engineering, 25(6), 5454–5497. https://doi.org/10.1007/s10664-020-09864-1
Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, Yi Zhang (2023). Sparks of Artificial General Intelligence: Early experiments with GPT-4. *arXiv.Org*. https://doi.org/10.48550/arxiv.2303.12712
Sharma, A., Chen, X., & Kushwaha, D. S. (2011). Natural language based component extraction from requirement engineering document and its complexity analysis. ACM Sigsoft Software Engineering Notes, 36(1), 1–14. https://doi.org/10.1145/1921532.1921547
Sillitto, H., J. Martin, D. McKinney, R. Griego, D. Dori, D. Krob, P. Godfrey, E. Arnold, and S. Jackson. 2019. Systems Engineering and System Definitions. San Diego, US-CA: INCOSE.
Srikanth Tammina. (2019). Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. *International Journal of Scientific and Research Publications*, *9*(10), 9420. https://doi.org/10.29322/ijsrp.9.10.2019.p9420
Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, & Luke Zettlemoyer. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv.Org*. https://doi.org/10.48550/arxiv.2305.14314
Majumder, A. (2020). *BERTSimilarity*. Retrieved from https://github.com/abhilash1910/BERTSimilarity
McKie, J. (2015). *PyMuPDF*. Retrieved from https://github.com/pymupdf/PyMuPDF
PyPDF2. (2007). PyPDF2 Team. Retrieved from https://github.com/py-pdf/pypdf
Rahman, M. M., & Finin, T. (2017). Deep Understanding of a Document's Structure. *BDCAT*, 63–73. https://doi.org/10.1145/3148055.3148080
Wu, X., Zhang, J., & Li, H. (2021). Text-to-Table: A New Way of Information Extraction. *Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2022.acl-long.180
Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, Hua Wu (2022). Unified Structure Generation for Universal Information Extraction. *Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.48550/arxiv.2203.12277
Zhong, X., Tang, J., & Yepes, A. J. (2019). PubLayNet: Largest Dataset Ever for Document Layout Analysis. *Null*. https://doi.org/10.1109/icdar.2019.00166

Biography

Ibrahim Ghanawi is an Engineer working at SysDICE GmbH since 2021. His work focuses on automating MBSE activities through AI capabilities, including in the fields of Natural Language Processing and Data Extraction. He has a diploma in Computer and Communications Engineering from the Lebanese University.

Mohammad Wissam Chami is a Model Based Software Developer working at SysDICE GmbH since 2021. He is skilled at using various programming languages and tools. Proficient in software development including plugin development, code generation, report generation and API development. He has a Bachelor of Engineering in Communications and Computer Networks Engineering.

Mohammad Chami is the founder and CEO of SysDICE GmbH, with an extensive career in the field of Model-Based Systems Engineering (MBSE). He is an MBSE expert with solid academic and industrial experience in modeling languages, processes, and the development and deployment of methods for systems modeling, as well as customizing its tools. In recent years, his focus has shifted towards the application of artificial intelligence (AI), specifically natural language processing and machine learning, in MBSE. Mohammad holds a Ph.D. in computer science from the Université Paul Sabatier Toulouse III and two master's degrees in Electronics and Mechatronics. He has been an active member of INCOSE since 2014.

Marko Coric is Lead Systems Engineer at Zühlke Engineering GmbH in Eschborn. The methodical development of systems has accompanied his entire professional and scientific career. In 2016, he obtained a doctorate in the field of methodical system development. Since completing his doctorate, he has been constantly expanding his specialist knowledge and adapting it to the challenges of everyday professional life. He firmly believes that the effective and efficient development of systems - whether medical, industrial or consumer products - requires pragmatic systems engineering approaches.

Nabil Abdoun is the Head of Research and Development (R&D) at SysDICE GmbH, with a distinguished career in the field of Artificial Intelligence (AI), Natural Language Processing (NLP) and Cybersecurity. Dr Abdoun has a Ph.D. in data and network security from Polytech Nantes, France. His work focuses on developing AI4MBSE track and his interests include Machine Learning, Systems Engineering, and System Security. Dr Abdoun has MS degrees in HRM, Network and Telecommunication, and CCE from the Lebanese University.