Machine Learning-based Classifier in the Analysis of Nuclear Power-Specific Requirements

Tapani Raunio, Ilpo Suominen, Santeri Myllynen and Rasmus Karell

April 6, 2021

Tapani Raunio*, Ilpo Suominen, Santeri Myllynen, and Rasmus Karell

Machine Learning-based Requirements Classifier in the Analysis of Nuclear Power Specific Requirements

Abstract: Typical nuclear power plant projects include a huge volume of stakeholder requirements to manage. These include requirements that are hard to interpret and error-prone to analyze and allocate to correct technical disciplines and processes. By utilizing machine learning in the analysis of nuclear power plant requirements, designers' decision making in classification and allocation of requirements could be facilitated and thus, errors reduced.

Fortum has developed a machine learning-based requirements classifier utilizing recent advantages in natural language processing (NLP) and integrated it with a requirements management system. The classifier categorizes project specific requirements into pre-defined categories.

Utilizing pre-trained language models allows training of a classifier with 12 categories with less than 2000 labelled requirements. Our model achieves 98% accuracy in classifying requirements from similar document sources as used in the model training phase.

The success of the classifier encourages to investigate other potential areas to utilize NLP. These are, among others, atomizing (i.e., splitting up) long, especially multi-category requirements, classifying requirements written in Finnish, requirement fulfillment assessment, identifying similar requirements, etc.

Keywords: machine learning, NLP, requirements engineering

***Corresponding Author: Tapani Raunio:** Fortum, E-mail: tapani.e.raunio@fortum.com

Ilpo Suominen: Fortum, E-mail: ilpo.suominen@fortum.com

Santeri Myllynen: Fortum, E-mail: santeri.myllynen@fortum.com

Rasmus Karell: Fortum, E-mail: rasmus.karell@fortum.com

1 Introduction

All safety-critical projects need requirements management to demonstrate that the outputs meet the requirements that have been set. Requirements management and engineering is a challenge in nuclear industry projects due to the large amount of requirements. Solely the Finnish regulatory guidance includes several thousand requirements, and requires to justify and apply relevant standards in the design. Considering all the levels in the requirements hierarchy, the amount of requirements can be from thousands to tens of thousands, and their content vary from very specific to very generic.

Successful projects must track, elaborate, and manage requirements from several sources. Requirements need to be analyzed and allocated to correct technical disciplines and processes as they affect design choices and process. A single misallocated requirement can lead to design changes and affect multiple technical disciplines and processes. Changes to the design in late phase of a project usually carry a hefty price tag, and pose a substantial risk of delay.

Fortum has developed a requirements classifier based on utilizing recent deep learning model architectures and natural language processing (NLP) to support requirements allocation. The requirements classifier is capable of suggesting one or multiple technical disciplines and processes for the experts' consideration. Motivation is to ensure that relevant technical disciplines are identified in the requirement analysis phase.

In this paper we describe our recent developments and suggest further research topics. This work is built upon Fortum's earlier studies on the topic [1, 2].

2 Use of Requirement Categories

2.1 Use Case for Requirements Classifier

Requirements driven design together with requirements management ensures that the design should meet all stakeholder needs and expectations.

Requirements come from many different sources, which are written in different styles, contain overlapping requirements, and in some cases contain even contradictory requirements. A requirement can be hard to interpret, and when analyzing large amount of requirements, the humans' limited ability to concentrate on a specific task causes errors.

The later possible mistakes in the design of a product are noticed, the greater the additional costs and delays are. Correcting errors in the late stage of a project causes redesign, which leads to larger costs.

Improving requirements analysis ability decreases the probability of design errors. Our use case is to use NLP based requirements classifier as a support tool in requirement allocation.

2.2 Integration with Requirements Management

Requirements management can be done with a wide range of tools. In our case we use Polarion® ALM^{TM} (later Polarion) Requirements Management System application.

We chose to integrate our requirements classifier with Polarion in order to automate the use of the classifier. From the user point of view adding a new requirement will lead to Polarion to suggest one or more requirement categories for the requirement. The final decision of correct requirement categories is always made by the user.

In the background Polarion and the classifier are running on separate servers, and communicating with web requests. In addition to the classifier server there is also a server to collect user saved requirement categories from Polarion, and a training server. Collecting categories saved by the user allows us to gather predicted and actual requirement categories. Training server is used to train new classifier models, and to analyze performance.

2.3 Requirement Categories

In this work we used 12 requirement categories, which consist of the following technical disciplines:

- Control Center Engineering
- Electrical Engineering
- HVAC Engineering
- I&C Engineering
- Process Engineering

and following process categories:

- Configuration Management
- Decommissioning
- Licensing
- Qualification
- Quality Management
- Requirements Management
- Verification & Validation

The selected requirement categories are based on our view of the main design life cycle related processes. At this point of development the categories do not cover all technical disciplines. For example mechanical, civil, and layout engineering among others are missing.

2.4 Data Collection

The goal of the data collection phase was to prepare a large set of requirements from multiple different sources for requirements classifier training and evaluation.

The collected dataset contains requirements from various sources such as Finnish Regulatory Guides on nuclear safety (YVL), IAEA requirements, ISO and IEC standards. Altogether requirements were collected from 40 different documents. All the requirements were in English.

Each requirement was allocated to one or more requirement categories mentioned in Section 2.3. Requirement allocation was done by discipline experts.

The data collection phase resulted in 2,819 requirements with each requirement in one or more categories from 12 possible categories. The median amount of requirements per category was 178.

3 Requirements Classifier

3.1 Natural Language Processing Text Classifier

Our requirements classifier is based on recent advances in NLP, and the readily available pre-trained state-of-the-art models. A pre-trained language model is already trained to "understand" text to a high degree, and we only need to teach the model classification task.

Pre-training enables the classifier training to focus on the requirement categories and enables classification when there is only a modest amount of labeled training data available.

The common approach in language model based NLP text classification is to transform input text to model's internal representation that provides features to the actual text classifier. Depending on the language model and the use case, the language model is fine-tuned as text classifier is trained or language model can be frozen as text classifier is trained.

Using a deep learning based language model allows fast model adaptation to new requirement data sets, because feature engineering is not needed.

3.2 Models

In this study the following representative language models are studied: BERT, DistilBERT, RoBERTa, XLM and XLNet [3-7].

BERT model has become the baseline for NLP projects and other chosen models expand or modify BERT various ways. DistilBERT provides comparable performance with improved speed and reduced resource consumption. RoBERTa uses vastly larger training set, Dynamic Masking Pattern and modified loss objective compared to BERT.

XLM model is similar to BERT but trained simultaneously on multiple languages leading to enhanced multilingual representations. XLNet uses generalized autoregressive approach with bidirectional context.

4 Experiments

We divided our 2,819 labelled requirements into following groups:

- Training dataset (n=1939)
- Validation dataset (n=414)
- Test dataset (n=390)
- Independent test dataset (n=76)

The training dataset was used to train the classifier part of the model, and fine-tune the pretrained language model for our classification task. The validation dataset was used in selecting model training hyperparameters. Model performance was assessed against the test dataset and the independent test dataset. The test dataset consists of requirements which were not used in the training or the validation dataset, but are from the same set of source documents. The independent test dataset consists of requirements from documents, which were not used in training or validation.

We used two different metrics to assess model performance: accuracy and match ratio. We define accuracy as the number of correctly labelled requirement categories divided by amount of requirements and requirement categories. Correctly predicted absences of categories are also counted in accuracy. In equation form

$$accuracy = \frac{\mathbf{1}}{\boldsymbol{n_r} \cdot \boldsymbol{n_c}} \sum_{\boldsymbol{s=1}}^{\boldsymbol{n_r}} \sum_{\boldsymbol{l=1}}^{\boldsymbol{n_c}} \mathbf{1}(\hat{\boldsymbol{y}}_{\boldsymbol{s,l}} = \boldsymbol{y}_{\boldsymbol{s,l}})$$

where $\boldsymbol{n_r}$ is the amount of requirements and $\boldsymbol{n_c}$ is the number of categories, $\hat{y}$ is the ground truth and y is the prediction.

Match ratio is defined as the ratio of completely correctly predicted requirements to all requirements. A requirement is completely correctly predicted when all its categories (and their absences) were correctly predicted. In equation form

$$Match\ ratio = \frac{\mathbf{1}}{\boldsymbol{n_r}} \sum_{\boldsymbol{s=1}}^{\boldsymbol{n_r}} \mathbf{1}(\hat{\boldsymbol{y}}_{\boldsymbol{s}} = \boldsymbol{y}_{\boldsymbol{s}})$$

We trained and evaluated model performance in 5 different model architectures: BERT, DistilBERT, RoBERTa, XLNet, and XLM. In all cases the models predicted probabilities to all requirements classes, and 0.5 probability was used as a threshold for label prediction.

5 Results

The results for the test dataset are shown in Table 5-1. The performance in the test dataset is excellent with accuracy around 97-98% and match ratio in BERT based models 82-86%. All BERT based models are close to each other while XLNet achieving similar results. Only XLM has noticeable lower match ratio than other models. BERT achieves the best performance in the test dataset.

Table 5-1. Model results for the test dataset.

	Accuracy	**Match ratio**
BERT	0.983	0.869
DistilBERT	0.981	0.849
RoBERTa	0.978	0.821
XLM	0.972	0.734
XLNet	0.980	0.815

Model performances in the independent test dataset in Table 5-2 achieve good accuracy, but match ratio is poor compared to the performance in the test dataset. XLNet model performs best in the independent test dataset. Since the size of the independent test dataset is small, a bigger portion of performance and performance differences between models can be due to chance.

Table 5-2. Model results for the independent test dataset.

	Accuracy	Match ratio
BERT	0.906	0.355
DistilBERT	0.902	0.289
RoBERTa	0.914	0.394
XLM	0.898	0.158
XLNet	0.934	0.434

The results for BERT, DistilBERT and XLNet have been published earlier in [2], but were included here for comparison purposes.

6 Discussion

There are several possible ways to improve model performance. Collecting more high-quality data will likely bring performance improvements, as the models can be trained with a larger dataset. Different fine-tuning strategies could improve performance noticeably in the independent test dataset. One possible improvement strategy would be to add more context into the requirements themselves by adding for example the subheadings and headings structure to the requirement.

The main limitation of the current models is the need to retrain classifier when the amount of requirement categories or their content change. It is relatively easy to add new categories if they do not change already collected and labelled data. Introducing a new category, which might be present in the already collected data would require checking and relabeling all so far collected data.

There are many potential and exciting applications of NLP. In this work we utilized NLP to classify requirements. Possible future research topics include classifying requirements written in Finnish, requirements fulfillment assessment, atomizing complex requirements to multiple simple requirements, and identifying almost identical requirements. NLP could also be useful in analyzing the consistency and identifying possible contradictions in the document collection of the project.

7 Conclusions

We have shown that recent and easily available NLP models can be used in requirements classification task with excellent accuracy of 98% when requirements are written in a similar style as the requirements used in the training. Although our requirements classifier was trained with nuclear industry related requirements, similar results are likely reachable in other domains.

The availability of pre-trained NLP models and the ease of use of libraries have reduced the barrier for experimenting and developing new exciting applications. The amount of collected and classified data needed for model training and test data is reasonable.

Our requirements classifier is ready to be used as a support tool in requirements engineering. Integration with requirements management software allows to automate the use of the classifier, and to collect more data in real projects. More data will likely allow to train better classifiers in the future.

References

[1] Myllynen S., M.Sc. thesis, Aalto University, Espoo, 2019.

[2] Myllynen S., Suominen I. , Raunio T., Karell R., Lahtinen J., Developing and Implementing Ai-Based Classifier for Requirements Engineering, ASME J. of Nuclear Rad Sci., 2021

[3] Devlin J., Chang M., Lee K., Toutanova K., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NAACL, 2019.

[4] Sanh V., Debut L., Chaumond J., Wolf T., DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, NeurIPS, 2019

[5] Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D. et al., RoBERTa: A Robustly Optimized BERT Pretraining Approach, arXiv, 2019

[6] Lample G., Conneau A. , Cross-lingual Language Model Pretraining, arXiv, 2019

[7] Yang Z., Dai Z., Yang Y., Carbonell J., Salakhutdinov R., Le Q. V., XLNet: Generalized Autoregressive Pretraining for Language Understanding, NeurIPS, 2019