IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 11, 2021, accepted September 19, 2021, date of publication September 24, 2021,
date of current version October 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3115659

SpaceTransformers: Language Modeling for Space Systems

AUDREY BERQUAND, PAUL DARM, AND ANNALISA RICCARDI
Intelligent Computational Engineering Laboratory, Department of Mechanical and Aerospace, University of Strathclyde, Glasgow G1 1XQ, U.K.

Corresponding author: Audrey Berquand (audrey.berquand@strath.ac.uk)

This work was supported in part by the European Space Agency through the Networking Partnership Initiative under Contract 4000123680/18/NL/MH, and in part by the Erasmus+ Student Mobility of Placement under Contract 2019-1-DE01-KA103-004732.

ABSTRACT The transformers architecture and transfer learning have radically modified the Natural Language Processing (NLP) landscape, enabling new applications in fields where open source labelled datasets are scarce. Space systems engineering is a field with limited access to large labelled corpora and a need for enhanced knowledge reuse of accumulated design data. Transformers models such as the Bidirectional Encoder Representations from Transformers (BERT) and the Robustly Optimised BERT Pretraining Approach (RoBERTa) are however trained on general corpora. To answer the need for domain-specific contextualised word embedding in the space field, we propose SpaceTransformers, a novel family of three models, SpaceBERT, SpaceRoBERTa and SpaceSciBERT, respectively further pre-trained from BERT, RoBERTa and SciBERT on our domain-specific corpus. We collect and label a new dataset of space systems concepts based on space standards. We fine-tune and compare our domain-specific models to their general counterparts on a domain-specific Concept Recognition (CR) task. Our study rightly demonstrates that the models further pre-trained on a space corpus outperform their respective baseline models in the Concept Recognition task, with SpaceRoBERTa achieving significant higher ranking overall.

INDEX TERMS Language model, transformers, space systems, concept recognition, requirements.

I. INTRODUCTION

In the past three years, the transformers architecture [1] and transfer learning [2] have profoundly impacted the Natural Language Processing (NLP) landscape. Transfer learning consists of two stages: (i) a pre-training phase in which contextualised word embeddings are learned through self-supervised training tasks on a large unlabelled corpus (for instance, Masked Language Model (MLM) and Next Sentence Prediction (NSP) [2]), and (ii) a second phase in which the pre-trained model is fine-tuned for a specific task [3]. The performance of the downstream NLP tasks are thus greatly improved with the knowledge transferred from the pre-trained models. Numerous studies presented the theoretical background and empirical proof of the positive impact of the pre-training and fine-tuning setting for downstream tasks [4], [5]. The BERT model, standing for Bidirectional Encoder Representations from Transformers, from Google AI Language [2] advanced the state-of-the-art (SOTA) performance on 11 NLP tasks. Transfer learning brings a decisive advantage for NLP applications, especially for domains where annotated corpora are scarce.

Space systems engineering is a field where access to large-scaled annotated data is limited. Yet, experts involved in the early stages of space mission design can spend up to 50% of their work time searching for heritage and design information [6]. The accumulated data explored by experts mostly consist of unstructured data: past design reports, books and journal publications. This information bottleneck can be reduced by implementing NLP and text mining solutions. Concept Recognition (CR) is a first essential step for the identification and extraction of domain-specific fundamental concepts, enabling the structuring of accumulated data via the construction of ontologies [7].

While pre-trained transformer models such as BERT [2] or RoBERTa, a Robustly Optimised BERT Pre-training Approach [8], are trained on general corpora, domain-specific models such as SciBERT [9] have proven to be more adapted to domain-specific downstream tasks. Pre-training language models from scratch is resource intensive, requiring large corpora (160 GB for RoBERTa [8]) and costly computational resources (7 days of training on

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco J. Garcia-Penalvo.

a Tensor Processing Unit (TPU) for SciBERT [9]). Instead, we propose to further pre-train the baseline models on our domain-specific corpus. We choose the BERT-Base, RoBERTa-Base and SciBERT-SciVocab models to build SpaceTransformers, a family of three models for space systems language modeling: SpaceBERT, SpaceRoBERTa and SpaceSciBERT. While models pre-trained on a general corpus learned contextualised word embeddings for a general or scientific English vocabulary, our further pre-training specialises these models in space systems engineering. The models performance are evaluated through a fine-tuning Concept Recognition (CR) task with a set of space systems terms annotated by hand by three human annotators. The contributions of this paper are summarised as follow:

1) We further pre-train and release SpaceTransformers, a novel open-source family of three models: SpaceBERT, SpaceRoBERTa and SpaceSciBERT further pre-trained from BERT, RoBERTa, and SciBERT on our space systems corpus.
2) We release a novel labelling scheme based on space standards and its corresponding hand-annotated dataset for Concept Recognition (CR) of space systems terms.
3) We provide, for the first time a thorough comparison of the performance of domain-specific models with respect to several baseline models on a classification task.
4) We demonstrate that further pre-training from RoBERTa-Base considerably improves the results on the downstream CR task for domain-specific language models.

The source code and domain-specific models are available at github.com/strath-ace/smart-nlp. All data underpinning this publication are openly available from the University of Strathclyde KnowledgeBase at https://doi.org/10.15129/8e1c3353-ccbe-4835-b4f9-bffd6b5e058b (further pre-training corpus) and https://doi.org/10.15129/3c19e737-9054-4892-8ee5-4c4c7f406410 (fine-tuning corpus and labelled concepts).

II. BACKGROUND AND RELATED WORK

A. TRANSFER LEARNING

The purpose of transfer learning is to first learn from an initial training objective, then apply it to a different target objective. Let s be an input sequence consisting of m words such that

$$s = (t_1, \ldots, t_m) \tag{1}$$

where t_i is the i^{th} word of the sequence. These tokens have a fixed initial embedding of dimension n, noted as x_i. The pre-training phase yields a contextualised embedding y_i of dimension d for each embedding x_i of a term t_i

$$f : \mathbb{R}^n \times \Theta_f \rightarrow \mathbb{R}^d, \quad f(x_i, \theta_f) = y_i \tag{2}$$

where $\theta_f \in \Theta_f$ represents a particular set of model parameters. In the pre-training phase, the model f is trained in a self-supervised fashion. In a second phase, the pre-trained model is fine-tuned for a specific task. The contextualised representations previously obtained are used as inputs to the model

$$g : \mathbb{R}^d \times \Theta_g \rightarrow \mathbb{R}^q, \quad g(y_i, \theta_g) = z_i \tag{3}$$

The output is a probability distribution through an identity or softmax activation function, configured by the parameters $\theta_g \in \Theta_g$ and of dimension q. The parametrisation of the fine-tuned model is thus configured by

$$\theta_{ft} = [\theta_f, \theta_g] \tag{4}$$

This framework has proven to be more efficient than training a task-specific model from scratch, requiring at least 10 times less task-specific data samples [2], [4]. The number of pre-training parameters, θ_f, is usually much higher than the number of fine-tuning parameters θ_g. For instance, the configuration of BERT-Base involves a $\theta_{f,BERT}$ of $110M$ parameters [2]. Thus, the training set required for fine-tuning is significantly smaller than for the pre-training, while avoiding over-fitting.

Finally, let $C(\cdot, \cdot)$ be the loss function for training a neural net (e.g. cross-entropy), then the cumulative empirical risk for minimising the loss in the fine-tuning setting is defined as:

$$\min_{\theta_f} C\left(f_{\theta_f}, X_f\right) + \min_{\theta_{ft}} C\left(g_{\theta_g}(f_{\theta_f}), Y_g\right) \tag{5}$$

where f_{θ_f} is the pre-trained model configured by θ_f parameters, $g_{\theta_g}(f_{\theta_f})$ is the fine-tuning model configured by θ_g parameters, and X_f, Y_g are respectively the pre-training and fine-tuning training sets.

B. DOMAIN-SPECIFIC LANGUAGE MODELS

There are three approaches found in the Literature to generate domain-specific language models: (i) a generic model is fine-tuned on a domain-specific task, (ii) a model is further pre-trained from a generic pre-trained model with a domain-specific corpus, or (iii) a model is trained from scratch on a domain-specific corpus.

Fine-tuning a pre-trained model for a domain-specific task is the quickest and easiest approach. In [10], the authors fine-tuned BERT-Base on a patent database for a classification task. Their model, patentBERT achieved better results than the previous SOTA method based on Convolutional Neural Network (CNN) and word vector embedding. Reference [11] presents a downstream application similar to our work. The authors fine-tuned BERT-Base on a CR task to identify concepts related to space systems engineering. To the best of our knowledge, their study is so far the only application of transfer learning in the space field. Their labelled dataset was however based on a single document, the NASA System Engineering Handbook [12] and they chose high-levels labels such as *event* or *location* whereas our labels cover all management, product assurance and engineering disciplines found in 126 space standards.

Pre-training from scratch or further pre-training on a domain-specific corpus enables the introduction of

domain-specific words embeddings in the language model, improving the performances on downstream domain-specific tasks. BioBERT [13] and VNLawBERT [14] were both further pre-trained from BERT-Base respectively with biomedical publications and a Vietnamese legal corpus. A clinical language model presented in [15] was further pre-trained from BERT-Base and from BioBERT. Both ClinicalBERT [16] and FinBERT [17] were trained from scratch on an architecture similar to BERT's with, respectively, a corpus of clinical notes and a large financial corpora. The benefits of either further pre-training or training from scratch on a domain-specific corpus have been largely proven by these studies as they all outperformed the original general language models on domain-specific tasks.

Further pre-training or training from scratch appears as a trade-off between (i) the available domain-specific corpus size, (ii) the available computational resources, and (iii) the fine-tuning performances sought-after. Training from scratch is resource intensive, it requires a large domain-specific corpus and heavy computational resources. Both BERT and SciBERT use a corpus of around 3B tokens. The training of BERT-Base was performed in 4 days on 4 cloud TPUs [2]. RoBERTa was trained in one day over 1024 V100 GPUs [8]. SciBERT took 7 days to train from scratch with a single TPU v3 with 3 cores [9]. In [18], a legal language model, LEGAL-BERT, is trained on a 12 GB corpus of legal texts, either from scratch or further pre-trained from BERT-Base. The authors found that both were valid approaches with similar results. Our training corpus has a similar size as [18] and we use a single NVIDIA V100 GPU with 16 cores to train our models. Based on these limitations, the decision was taken to further pre-train our domain-specific models rather than train them from scratch. The methods mentioned in this Literature Review are summarised in Table 1.

TABLE 1. Methodologies comparison. PT stands for pre-training from scratch, FPT for further pre-training, and F for fine-tuning.

	Source	Domain	Approach	Baseline Model	Training Corpus
Baseline	BERT-Base [2]	General	PT	-	16 GB (3.3 Billion tokens)
	RoBERTa-Base [8]	General	PT	-	160 GB
	SciBERT [9]	Science	PT (SciVocab)	-	3.17 Billion tokens (≈ 15.4 GB)
Domain-Specific Models	patentBERT [10]	Legal	F	BERT	-
	SEVA [11]	Space	F	BERT	-
	ClinicalBERT [16]	Clinical	PT	-	2 Million notes
	FinBERT [17]	Finance	PT	-	61 GB
	LEGAL-BERT [18]	Legal	PT FPT	- BERT	12 GB
	BioBERT [13]	Biomedical	FPT	BERT	18 Billion tokens (≈ 87 GB)
	VNLawBERT [14]	Legal	FPT	BERT	320 MB
	Clinical BERT [15]	Clinical	FPT	BERT BioBERT	2 Million notes
	This paper	Space	FPT	BERT RoBERTa SciBERT	14.3 GB

C. CONCEPT RECOGNITION FOR SPACE SYSTEMS

CR is a NLP task used to identify and classify terms of interest from text. It is a word-level annotation exercise. For instance CR in the clinical domain annotates labels associated with general terms, including, in the analysis of patient data, terms such as "*treatments*", "*findings*", and "*problems*" [19]–[21]. Similarly to the clinical domain, CR for space systems engineering includes generic terms, describing, for instance, the interface between engineering and management [11]. Therefore, concepts can be loosely defined as sequences that represent a specific cognitive construct in their domain [22]. In the context of systems engineering, these concepts can be "*engineering unit*", "*system architecture*" or "*system analysis*", labelled as examples for the label "*system concepts*" in [11]. One can assume that in systems engineering the concept "*system*" almost exclusively stands for the technical assembly of interconnected items or devices of a satellite or spacecraft, in comparison to generic text where "*system*" could have different meanings based on context. In general, ambiguity depends on the target domain as well as on the level of granularity in the annotation scheme defining the level of abstraction. For instance, labels such as "*tasks*", "*processes*", and "*materials*" were used for constructing a scientific knowledge graph in [23]. These labels can be applied to multiple scientific domains such as computer science, biology, and mathematics, and thus have a low level of granularity with a high chance of ambiguity as the meaning of a concept varies in function of the scientific field. Nevertheless for the purpose of comparing scientific publications based on their intrinsic concepts, this level of granularity is considered as sufficient [23]. Thus, the necessary level of granularity in the annotation scheme for CR depends on the later application, target domain, and their tolerated level of ambiguity.

Different approaches for CR applications exist. Rule-based and pattern matching systems leverage hand-crafted rules on the text and its linguistic features to extract concepts as shown in [24]. Alternatively, other methods are based on supervised Machine Learning (ML) methods, trained from example inputs and their expected outcomes. Linguistic feature-based ML systems such as support-vector-machines, decision trees, and conditional random fields used to be the preferred methods for CR [20]. However, in the last years, these were increasingly replaced by deep learning approaches using word embedding as input features [19], [23]. Language models and transfer learning have recently significantly contributed to this field. Transfer learning increases the performances of CR applications, as seen in [25], [26], requiring a smaller labelled dataset than training from scratch. Furthermore, the contextualised representation contributes to recognising and differentiating concepts based on their context, thus increasing the accuracy of the model predictions.

III. CORPORA

The study involves two corpora:

1) A further pre-training corpus: a 14.3 GB collection of unstructured documents related to space systems, acquired from heterogeneous sources.
2) A fine-tuning corpus: 28,763 textual requirements extracted from European Cooperation for Space Standardisation (ECSS) standards.

A. FURTHER PRE-TRAINING CORPUS

The training corpus is a collection of 5,266 unstructured documents including books, publication abstracts, and Wikipedia pages. These documents were manually gathered. They were chosen as they represent the typical information sources used by space systems engineers. The books cover most of the fields of space mission design, and are publicly available. The abstracts were extracted from papers published in three peer-reviewed journals: the Acta Astronautica, Advances in Space Research, and the Aerospace Science and Technology journals. All papers were published between 2017 and 2019 included, and therefore describe recent work. Using the abstracts of the publications was found to yield better results than using the full journal publications documents. The reason is most likely that papers include mathematical notations, figures and tables which introduce noise. The Wikipedia webpages were scraped and manually filtered using the hyperlinks connecting pages to the spacecraft design webpage. Table 2 provides statistics on the training corpus. The sentences are mainly extracted from books (70%), then from publication abstracts (17,6%) and Wikipedia (12,4%). This distribution reflects the language complexity of these different sources.

TABLE 2. Statistics of the further pre-training corpus.

Sources	Publications Abstracts	Books	Wikipedia Webpages	All
Number of documents	4,953	40	273	5,266
Number of sentences	37,957	152,143	26,942	217,042
Average number of tokens per sentences	27,1	24,4	24,2	24,8

B. FINE-TUNING CORPUS

The fine-tuning corpus consists of annotated requirements extracted from ECSS standards. The latter is an initiative launched by the European Space Agency (ESA) in 1993 to define a coherent and single set of standards for all European space activities [27]. 28,763 requirements are collected from 126 single standards as shown in Table 3. The ECSS standards are split into three main branches under an overhead branch called *System*: *Management*, *Product assurance* and *Engineering*, covering the design and implementation of the standards and requirements. Each requirement briefly describes a regulatory provision to be complied with in the form of "what to do" in a customer - supplier context [28]. Because of the intent of using them in an obligating contract, the requirements are written in a clear and unambiguous language. Additionally, the average number of tokens per requirement is similar for all branches.

TABLE 3. Statistics of the fine-tuning corpus.

Allocation	Management	Product Assurance	Engineering	All
Number of standards	5	60	61	126
Number of requirements	558	9,338	18,867	28,763
Number of tokens per requirement	29	29	31	30

For the fine-tuning, we used requirements from the three branches. Focusing on just the majority branch *Engineering* would not be feasible as the standards are to be used in conjunction with each other and not as single documents. For instance, the topic "Software" is covered by two standards belonging to the *Engineering* and the *Product assurance* branches. Nevertheless, there is an effort to avoid duplication of content in requirements with the ideal situation that each requirement is unique [29].

IV. METHODOLOGY

The SpaceBERT, SpaceRoBERTa and SpaceSciBERT models are respectively further pre-trained from BERT-Base, RoBERTa-Base, and SciBERT-SciVocab. The pre-trained and further pre-trained models are fine-tuned on a domain-specific CR task. The methodology is summarised in Figure 1.

A. FURTHER PRE-TRAINING

Further pre-training a model

$$f : \mathbb{R}^n \times \Theta_f \rightarrow \mathbb{R}^d, \tag{6}$$

means that in the pre-training phase, instead of randomly initialising the weights θ_f, the weights values of a baseline model such as BERT, RoBERTa or SciBERT are reused. Hence the weights θ_f for the three further pre-training tasks are initialised with the following set of weights

$$\theta_{f,0} = \theta_{f,BERT}, \tag{7}$$
$$\theta_{f,0} = \theta_{f,RoBERTa}, \tag{8}$$
$$\theta_{f,0} = \theta_{f,SciBERT} \tag{9}$$

where $\theta_{f,BERT}$, $\theta_{f,RoBERTa}$ and $\theta_{f,SciBERT}$ are respectively the set of weights of the pre-trained models BERT, RoBERTa and SciBERT. Weights initialisation from a pre-trained model also implies the reuse of the original model vocabulary. The authors of the SciBERT model [9] observed an average improvement of only +0.76 F1 score on biomedical tasks when using their domain-specific vocabulary.

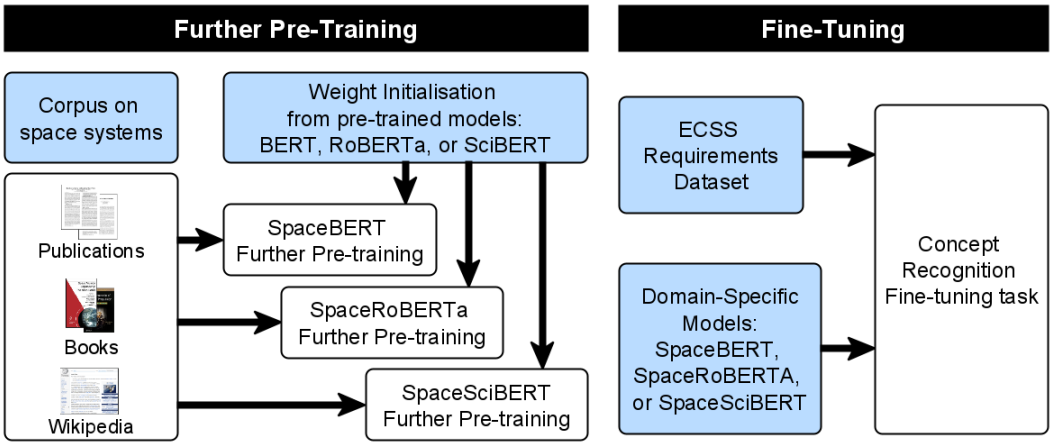


FIGURE 1. Overview of the further training and fine-tuning methodology with SpaceBERT, SpaceRoBERTa, and SpaceSciBERT.

They concluded that training with a domain-specific corpus had more impact than using a domain-specific vocabulary. A study similar to ours, BioBERT [13], chose to rely on the BERT-Base vocabulary. The authors assessed that since the Word Piece tokenization used to build the BERT vocabulary reduces out-of-vocabulary issues it was fit to represent and fine-tune their domain-specific terms. An alternative to training from scratch with a domain-specific corpus is to replace "Unused" tokens in the vocabulary with domain-specific words. To assess if a modification of the original vocabulary was necessary, we extracted the top thousand most frequent words from our domain-specific corpus and compared our frequency-based lexicon to the vocabulary of BERT-Base-uncased, RoBERTa-Base, and SciVocab-uncased. The top 10 most frequent words in our frequency-based lexicon are: "*satellite*", "*system*", "*orbit*", "*space*", "*spacecraft*", "*data*", "*time*", "*mission*", "*model*", and "*control*". Out of our frequency-based lexicon, 87, 8% of the words were already included in the BERT-Base-uncased vocabulary, 88, 8% in the RoBERTa-Base vocabulary, 89, 9% in the SciVocab. Within these 1000 words, the 10% most frequent words were already included in all three vocabularies. Table 4 gives a sample of the words not found in the generic models vocabularies. As the amount of domain-specific terms not covered by the original vocabularies was negligible, we decided to re-use the vocabularies and tokenizers of the models we were further training on.

The configuration and pre-trained weights of the BERT-Base, RoBERTa-Base and SciBERT models are accessed through the HuggingFace library and their Python Transformers library [30]. For each model the pre-training weights and hyperparameters are thus initialised from one of the three baseline models with the exception of the batch size and maximum sequence length. The batch size is set to 256, as for RoBERTa [8], with a gradient accumulation step of 16. The maximum sequence length of the input is set to 512 as defined in BERT [2]. The models are further pre-trained for 70 epochs on one NVIDIA V100 GPU hosted on the ARCHIE-WeST High Performance computer. The further pre-training corpus is split between a training and a testing set, based on the classic 80%/20% partition.

TABLE 4. Sample of terms not found in the generic models, words in bold are missing from more than one vocabulary.

Vocabulary	BERT-Base-uncased	RoBERTa-Base	SciVocab
Number of terms not found in baseline vocab.	123	112	101
Top 5	subsystem, **thruster**, **propellant**, **perturbation**, **telemetry**	**thruster**, **propellant**, **perturbation**, **telemetry**, **actuator**	rocket, **thruster**, **propellant**, lunar, **telemetry**

B. REQUIREMENTS LABELLING

For the fine-tuning of the pre-trained models, the corpus presented in section III-B was used as a basis for the annotated dataset. The requirements are written in a precise and brief manner, with a high density of concepts relevant to space systems, making them useful for generating a CR dataset in this domain. An annotation scheme was carefully designed to cover the whole spectrum of the ECSS standards, creating labels for each of the three main branches: *Management*, *Product assurance* and *Engineering*. The labels were constructed from domain-experience of three human annotators as well as with the help of online available taxonomies in the space domain such as the ESA Technology tree [31], the ESA Product tree [32] and the NASA taxonomy viewer.[1] 18 labels were eventually defined for the annotation scheme. The complete description for each label is found at github.com/strath-ace/smart-nlp. Table 5 summarises the annotation

[1] https://techport.nasa.gov/view/taxonomy

TABLE 5. Annotation scheme summary.

Label	Short description	*Examples*
Management		
Project documentation	Project deliverables	*Certificate of conformity, PCB definition dossier, final review team report*
Project scope	Functions, characteristics, and goals of the mission	*mission phases, project requirements, product functionality*
System engineering	Design, components and functions of a system solution	*dynamic architecture design, system level considerations,*
Product assurance		
Nonconformance	Non-fulfilment of a requirement	*explosion, material degradation, cuts, abrasions*
Quality control	Compliance with requirements and specifications	*acceptance tests, evaluation report, unit level testing*
Safety & risk control	Dependability, availability, maintainability, and safety	*safety-approved procedure, worst-cases, emergency controls*
Engineering		
Cleanliness	Contamination and sterilisation	*system cleanliness, contamination control plan, particle counter*
Communication	Communication and navigation infrastructure for telemetry/telecommand (TM/TC)	*telecommand packet, Link budget, message subtype 28*
Guidance Navigation & Control (GN&C)	Design and implementation of control subsystem, analysis and definition of trajectory	*star sensor, natural perigee rise, apogee fall*
Materials & EEE	Electrical, electronic and electro-mechanical (EEE) Components, materials	*Printed Circuit Board, sandwich items, fastener*
Measurement	Physical units	*30 J, 60 mW*
On-Board Data Handling (OBDH)	Data management, data acquisition, data storage, on-board networking and network management	*data-sending lane, DATA OUT signal, Distribution Transfer Descriptor*
Parameter	Generic characteristic	*supplier performance, track width, manufacturing tolerances*
Power	Power subsystem architecture, energy storage, power generation; distribution; and conditioning	*nuclear-energy sources, RTGs, output short circuit, power-energy resources*
Propulsion	Generation of forces and torques to change velocity and orientation of S/C	*thruster generated plasma, sloshing analysis*
Space environment	Effects and environmental conditions governing the space environment	*displacement damage, secondary protons, electron-bremsstrahlung*
Structure & mechanisms	Structural and mechanical subsystem, mechanism subsystem devices	*satellite mechanical structure, static unit load, attachment devices*
Thermal	Thermal management	*thermo-optical properties measurement, heat pipe, two- phases heat transport equipment*

scheme, providing a short description and examples for each label.

The single requirements were annotated with the commercial software tool Prodigy from the software company explosion.ai.[2] To facilitate the annotation process, requirements addressing similar topics were annotated simultaneously. The process was repeated for all topics, ensuring that similar numbers of requirements were selected so that the resulting dataset would be balanced and cover the full scope of the ECSS standards. The annotation process was considered done once the performance of the CR classifier were within an acceptable accuracy. Eventually, 882 requirements were annotated. Each annotator labelled the whole fine-tuning corpus independently. These results were then compared, showing a high level of inter-annotator agreement of 96.5%. Discrepancies between the three annotators were discussed and removed from the final set. The resulting numbers of annotated concepts present in the final dataset are shown in Table 6. The number of unique concepts found per label, as well as the ratio of unique concepts to the total number of concepts, called non-overlapping, are also displayed.

[2]https://prodi.gy/

TABLE 6. Summary of annotated concepts per label.

Labels	Number of concepts	Number of uniques	Ratio non-overlapping
Quality control	529	366	0.69
Space environment	518	392	0.76
Parameter	433	327	0.76
OBDH	410	317	0.773
System engineering	331	192	0.58
Measurement	301	260	0.86
Power	287	234	0.815
Materials & EEEs	275	203	0.74
Structure & mechanisms	253	223	0.88
Safety & Risk control	225	188	0.84
GN&C	206	164	0.8
Project scope	203	153	0.75
Communication	191	152	0.8
Thermal	185	139	0.75
Project documentation	154	122	0.79
Propulsion	151	118	0.78
Nonconformance	118	88	0.75
Cleanliness	91	64	0.70
Sum / mean*	**5447**	**4112**	**0.78***

C. FINE-TUNING FOR CONCEPT RECOGNITION

The Python Transformers library from HuggingFace [30] was used to load the pre-trained and further pre-trained models. For CR, a linear layer is added as output layer with a softmax activation function. The models were trained three times with a 10-fold, 80% to 20% split, cross validation. The split size was established from the mean ratio of non-overlapping samples, which is slightly below with 78%, as shown on Table 6. Another assumption for the training was to reinitialise the weights of the final layer if the fine-tuning resulted in a failed run for the fold. This is in accordance with previous studies, which stated that the random initialisation of the fine-tuning layers can have a significance influence on the fine-tuning results in computer vision [33] as well as NLP [34]. A failed run was defined as when the validation accuracy stayed below classifying all examples with the majority class, classifying every word as a non-concept [35].

Further hyperparameters for the fine-tuning were a linear decreasing learning rate and a batch size of 16. The models were trained for up to 10 epochs. To compare the models' predictions, the results of the epoch with the lowest validation loss for each respective fold were taken. One benefit of the further pre-training was already observed during fine-tuning. In comparison to RoBERTa with three failed runs overall, SpaceRoBERTa did not fail any.

V. RESULTS

A. MODELS SELECTION

During the trial and error phase, we experimented with uncased and cased vocabularies and various batch sizes. Further pre-training on uncased vocabulary yielded better results than cased vocabulary. This was to be expected as our labelled concepts are not named entities and thus casing is not relevant to our application. We also found that a higher batch size of 256 yielded better results than lower batch sizes of 16 or 32.

The models are further pre-trained for 70 epochs which is enough to achieve the convergence of the evaluation perplexity as shown in Figure 2. Perplexity is a common metrics for evaluating language models. It quantifies how well a model reduces the uncertainty in the prediction of the language in a tokenized sequence of text s. Perplexity PPL is derived from the cross-entropy H and is defined in [36] as:

$$PPL = 2^{H_p(s)} \tag{10}$$

with

$$H_p(s) = \frac{1}{m} \log_2 \frac{1}{P(s)} \tag{11}$$

where m is the number of words in the sequence s, $P(s)$ is the probability of the sequence of words provided by the model, $H_p(s)$ the cross-entropy of the text in relation to the model, and finally PPL the perplexity of the model.

We chose to retain the SpaceBERT model trained for 60 epochs, the SpaceRoBERTa model trained for 57 epochs, and the SpaceSciBERT trained for 54 epochs. These models either correspond to the start of the perplexity convergence or to a local minimum close to convergence. Although of disparate initial configuration and pre-training corpus, these models interestingly take a similar number of further pre-training epochs to converge.

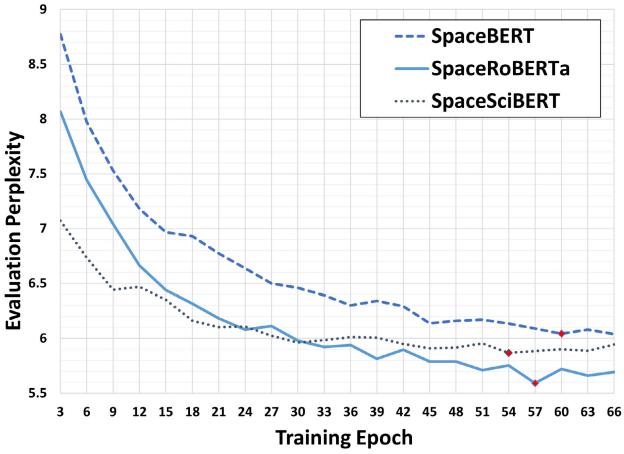


FIGURE 2. Evolution of the evaluation perplexity in function of the number of further pre-training epochs.

B. CONCEPT RECOGNITION RESULTS

Figure 3 displays the evolution of the validation loss for all models with respect to the number of fine-tuning epochs. The validation loss curves have a parabola-like shape reaching a minimum after a certain number of epochs. When comparing the minimums of each model, the validation loss appears to be the lowest for SpaceRoBERTa and the highest for BERT. While SpaceSciBERT and SciBERT have similar validation losses, SpaceRoBERTa, and SpaceBERT demonstrate significant improvements with respect to their respective baseline models. Although the results were averaged over 30 folds, the standard deviation for the validation loss is still high. Former studies [34], [35] reported similar issues for comparable dataset sizes.

The CR F1 scores for all 6 models and 18 labels are reported in Table 7. The results were computed from the epochs with the lowest average validation loss, averaged over all 30 folds. The standard deviation is provided along with

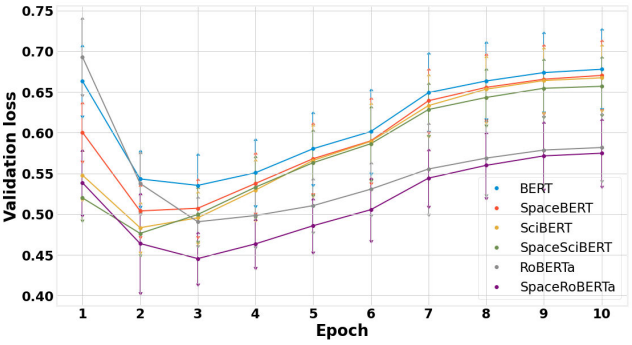


FIGURE 3. Evolution of the validation loss in function of the number of fine-tuning epochs.

the F1 score. The *weighted* label represents the averaged F1 score over all the labels weighted by the number of examples in the validation set, and is defined as:

$$weighted = \frac{1}{\sum_{l \in L} n_{\hat{y}_l}} \sum_{l \in L} n_{\hat{y}_l} F_1(y_l, \hat{y}_l) \qquad (12)$$

where l is one label from the set L of all labels, $\hat{y}_l$ is the set of true samples for label l, y_l is the set of predicted samples for label l, $F_1(y_l, \hat{y}_l)$ is the F1 score calculated for label l, and $n_{\hat{y}_l}$ is the number of true samples for label l.

Considering only this weighted F1 score, SpaceRoBERTa clearly outperforms the other models, followed by SpaceSciBERT. BERT and RoBERTa obtain the lowest scores. SpaceRoBERTa ranks the highest on several labels. As shown on Table 7, the labels, displaying the most significant improvements compared to the baseline of BERT, are *GN&C* with a 7.8% improvement, then *Space environment* with 4.5%, followed by *Thermal* with around 4% improvement, and *Structure & mechanism* 3.8%. SpaceSciBERT also substantially improves the score of the *Communication* and *OBDH* labels, respectively by 12% and 4%, compared to BERT.

Altogether, the reported F1 scores are consistent with the observed validation loss trends, with SpaceRoBERTa leading the F1 score table and the further pre-trained models outperforming their baselines. The standard deviations of the single scores are still generally high, usually exceeding the achieved improvement between the baseline and the further pre-trained models. Therefore, statistical tests are conducted and summarised in section V-C to evaluate the statistical significance of the results.

To fully assess the impact of the further pre-training with a domain-specific corpus, the scores of the baseline models are compared to their respective space variant in Figure 4. SpaceRoBERTa again displays the most significant improvements compared to its baseline model RoBERTa. All three domain-specific models show substantial improvements for the *Propulsion*, *Space environment*, *Structure & mechanisms*, *Communication*, *GN&C*, and *OBDH* labels. These labels corresponds to the main engineering disciplines of a spacecraft subsystems. However the score of more general labels such as *Safety & risk control*, *Nonconformance*, and *Quality control* were either unaffected or slightly deteriorated by the further pre-training. These labels all belong to the ECSS branch of *Product assurance*. For the remaining labels, no clear trend can be inferred as the further pre-training resulted either in an improvement or a deterioration of performances depending on the model used.

A more thorough investigation is conducted for the SpaceRoBERTa model as it achieved the highest performance. Figure 5 displays the confusion matrix for the SpaceRoBERTa model. The majority of samples are concentrated on the diagonal, thus predictions are predominantly accurate. A few incorrect classifications occur between the *OBDH* and *Communication* labels, indicating a lack of clear boundaries between the two topics.

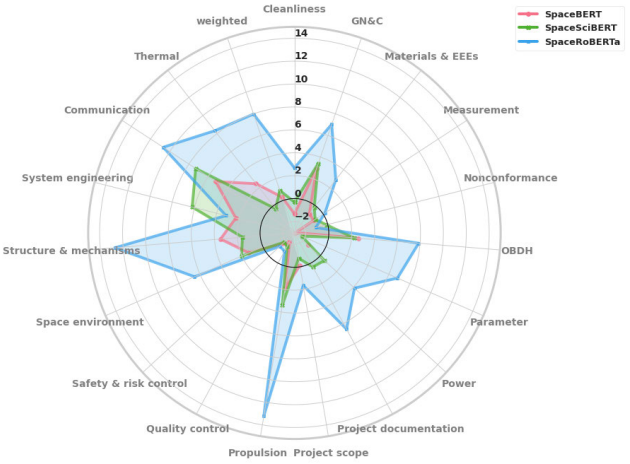


FIGURE 4. Variations between the performance of the baseline models and respective further pre-trained space models.

The annotated requirement shown in Figure 6 illustrates this overlap. SpaceRoBERTa wrongly associates the concepts found in this requirement to the *Communication* label instead of the *OBDH* label as they were manually assigned to. These concepts, including *communication frame* and *command word*, actually fall under the domain of signal processing and can be used both in a communication or data handling context. The requirement was here extracted from a standard related to data handling. This information is however hidden from the model and therefore cannot be used to guide it. The ambiguity of these terms were already highlighted by the human annotators.

Figure 7 quantifies the number of new concepts not seen by the model during training but found in the validation set, demonstrating the ability of the model to generalise over the training set and discover new concepts in unevaluated samples. The prediction of the model was compared for one fold to a simple look-up approach. The latter method identifies concepts present in both training and validation sets. As seen in Figure 7, the prediction with the fine-tuned model achieves substantially better results than the look-up approach. Out of 844 unique concepts, 690 were recognised exactly by the model and 78 concepts were partly recognised. For partial recognition, the span was either too long or too short. For instance, the concept *50W resistors*, corresponding to two labelled concepts *50W* and *resistor* were merged by the model. The concept *flight production* was extracted by the model while the full labeled concept was *proto-flight production*. Alternatively, the look-up approach resulted in only 170 complete and 187 partial matches.

C. STATISTICAL TESTS

The results obtained have been statistically analysed with the Friedman pre-hoc and the Bonferroni-Dunn and Nemenyi post-hoc tests. To determine the statistical significance of the F1 score of each method with respect to the labels set, a non-parametric Friedman test was completed with the ranking of the F1 score of the best model as the test variable.

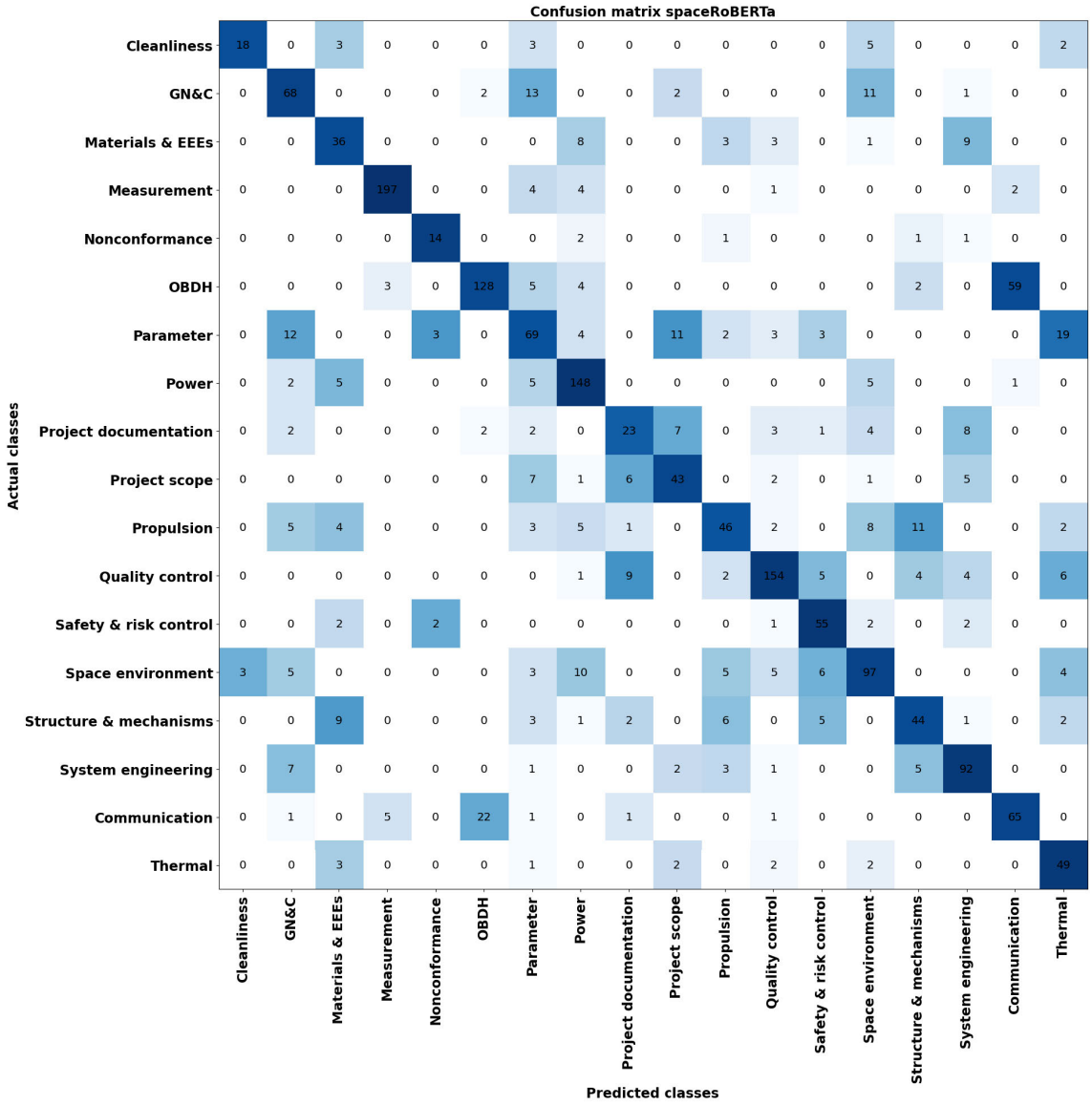


FIGURE 5. Confusion matrix of the fine-tuned SpaceRoBERTa model (the majority class "non-concept" is excluded).

The Friedman test shows that the proposed method is statistically significant at a level of 5% as the confidence interval is $C_0 = (0, F_5 = 2.322)$ and the F-distribution statistical values is $F^* = 6.330 \notin C_0$. Consequently the Friedman test rejects the null-hypothesis that all models perform equally well in mean ranking. Based on this rejection the Nemenyi post-hoc is completed to compare the performances of the different models. The difference in ranking, as resulting from the Nememyi tests can be observed in Figure 8, for $\alpha = 0.05$. The results of the Bonferroni-Dunn test for $\alpha = 0.05$ are reported in Table 7. From the results of both tests it can be concluded that SpaceRoBERTa has a significant higher ranking than all the other methods and RoBERTa, its baseline, the lowest one. The remaining methods, BERT, SciBERT and their space counterpart instead, have not a significant difference in mean ranking.

VI. DISCUSSION AND FUTURE WORK

The weighted F1 scores demonstrate that the domain-specific models outperformed their respective baseline models. SpaceRoBERTa benefited the most from the further pre-training with an increase of 8% F1 score with respect to RoBERTa. SpaceBERT and SpaceSciBERT have less significant improvements, respectively displaying increases of 0.3% and 0.85%. Both SpaceSciBERT and SciBERT outperformed SpaceBERT and BERT proving that the scientific pre-training gave an additional advantage to training from a general model. The decisive advantage came from

TABLE 7. Results for the F1 scores of 30-fold cross-validation for each model and each label. The best score for each label is highlighted with a grey background. The standard deviation is presented for each label behind the respective F1 score.

Label	BERT	SpaceBERT	SciBERT	SpaceSciBERT	RoBERTa	SpaceRoBERTa
Cleanliness	$0.709_{0.101}$	$0.699_{0.108}$	$0.71_{0.109}$	$0.708_{0.127}$	$0.703_{0.138}$	$\mathbf{0.722_{0.105}}$
GN&C	$0.703_{0.082}$	$0.723_{0.064}$	$0.715_{0.068}$	$0.739_{0.055}$	$0.708_{0.075}$	$\mathbf{0.758_{0.071}}$
Materials & EEEs	$0.664_{0.083}$	$0.659_{0.073}$	$0.655_{0.086}$	$0.654_{0.062}$	$0.647_{0.071}$	$\mathbf{0.665_{0.069}}$
Measurement	$0.888_{0.024}$	$0.879_{0.024}$	$0.883_{0.033}$	$0.875_{0.034}$	$0.889_{0.03}$	$\mathbf{0.89_{0.026}}$
Nonconformance	$\mathbf{0.558_{0.079}}$	$0.542_{0.105}$	$0.517_{0.097}$	$0.517_{0.072}$	$0.543_{0.126}$	$0.537_{0.098}$
OBDH	$0.736_{0.054}$	$0.755_{0.063}$	$0.751_{0.051}$	$\mathbf{0.767_{0.057}}$	$0.706_{0.062}$	$0.761_{0.054}$
Parameter	$0.521_{0.046}$	$0.504_{0.056}$	$0.528_{0.054}$	$0.516_{0.048}$	$0.505_{0.04}$	$\mathbf{0.539_{0.055}}$
Power	$0.816_{0.042}$	$0.805_{0.058}$	$0.824_{0.041}$	$\mathbf{0.829_{0.04}}$	$0.786_{0.076}$	$0.819_{0.044}$
Project documentation	$0.508_{0.058}$	$0.487_{0.072}$	$0.489_{0.081}$	$0.491_{0.076}$	$0.479_{0.1}$	$\mathbf{0.511_{0.073}}$
Project scope	$\mathbf{0.624_{0.052}}$	$0.623_{0.063}$	$0.621_{0.063}$	$0.616_{0.062}$	$0.607_{0.069}$	$0.617_{0.06}$
Propulsion	$0.699_{0.065}$	$0.712_{0.057}$	$0.684_{0.063}$	$0.707_{0.057}$	$0.637_{0.066}$	$\mathbf{0.722_{0.053}}$
Quality control	$\mathbf{0.734_{0.049}}$	$0.718_{0.046}$	$\mathbf{0.734_{0.045}}$	$0.722_{0.048}$	$0.731_{0.053}$	$0.723_{0.049}$
Safety & risk control	$0.689_{0.047}$	$0.678_{0.052}$	$0.688_{0.06}$	$0.676_{0.051}$	$\mathbf{0.701_{0.063}}$	$0.692_{0.067}$
Space environment	$0.74_{0.068}$	$0.75_{0.053}$	$0.757_{0.055}$	$0.772_{0.049}$	$0.725_{0.11}$	$\mathbf{0.773_{0.056}}$
Structure & mechanisms	$0.542_{0.084}$	$0.56_{0.075}$	$0.547_{0.092}$	$0.556_{0.084}$	$0.499_{0.113}$	$0.563_{0.087}$
System engineering	$0.617_{0.061}$	$0.631_{0.06}$	$0.592_{0.064}$	$0.629_{0.062}$	$0.61_{0.075}$	$\mathbf{0.63_{0.064}}$
Communication	$0.644_{0.084}$	$0.677_{0.068}$	$0.672_{0.094}$	$\mathbf{0.721_{0.059}}$	$0.616_{0.134}$	$0.682_{0.108}$
Thermal	$0.742_{0.045}$	$0.76_{0.063}$	$0.758_{0.046}$	$0.756_{0.054}$	$0.712_{0.108}$	$\mathbf{0.772_{0.055}}$
weighted	$0.699_{0.019}$	$0.701_{0.024}$	$0.703_{0.02}$	$0.709_{0.019}$	$0.662_{0.114}$	$\mathbf{0.715_{0.029}}$
Control method	Bonferroni-Dunn test					
SpaceRoBERTa	1.639•	1.833•	1.778•	1.694•	3.055•	-

Bonferroni-Dunn test $CD_{\alpha=0.05} = 1.606$
• Statistically difference with $\alpha = 0.05$

combining our domain-specific training corpus with the alternative pre-training architecture and tokenizer of RoBERTa. Indeed, the latter model is pre-trained on a single Masked Language Model (MLM) task [8] where the model must predict randomly hidden tokens whereas the BERT-based models are also trained on a Next Sentence Prediction (NSP) task [2], [9]. The statistical analysis and Bonferroni-Dunn test, ignoring the number of labels in the evaluation set unlike the weighted F1 score, demonstrated that there is no significant difference between SpaceBERT, SpaceSciBERT and their baseline counterpart. The Bonferroni-Dunn test however confirmed the significant higher ranking of SpaceRoBERTA.

Labels covering more common concepts such as *Nonconformance*, *Project Scope*, and *Quality Control* benefited less from the domain-specific training. Domain-specific labels such as *Propulsion*, *Structure & Mechanisms*, and *Communication* however saw their F1 score significantly increased for all space models. These results were obtained for one fine-tuning task. When fine-tuning for another task it is recommended to not discard SpaceSciBERT nor SpaceBERT as different models might be more adapted to different applications.

In future work, other pre-training tasks, beyond MLM and NSP, could be explored as in [37] where a domain-specific model was trained on four different tasks. This is a resource intensive approach requiring additional computational power and a larger training set. To improve the performances over ambiguous concepts that could be belong to several engineering disciplines, information should be integrated about the original document the requirements were extracted from. Related to the fine-tuning, the comparison could be extended to additional downstream tasks to further compare the performances of SpaceRoBERTa, SpaceSciBERT and SpaceBERT.

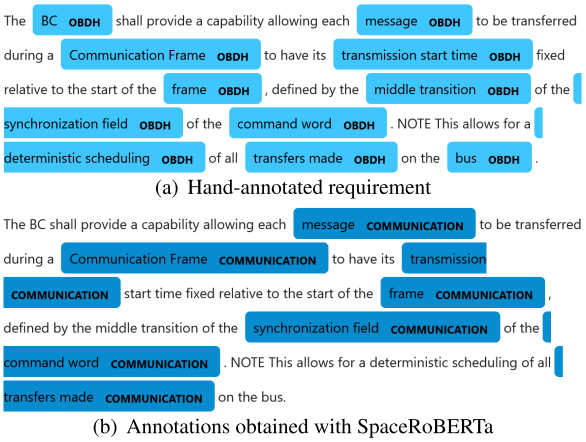


FIGURE 6. Comparison of manual annotation and model prediction.

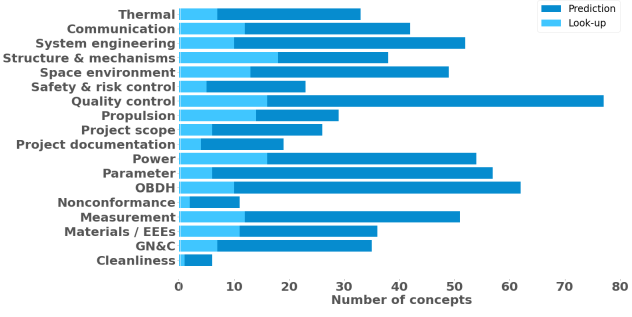


FIGURE 7. Number of unique concepts detected by the SpaceRoBERTa model, compared to a simple look-up approach.

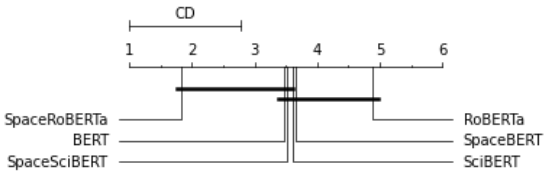


FIGURE 8. Nemenyi CD diagram comparing the generalization F1 score rankings of the different methods ($\alpha = 0.05$).

CR can as well support additional text mining operations on the ECSS standards. Standards contain key information on space systems, and they are highly correlated. Thus, a follow-up task could be to associate similar requirements based on common concepts. This application could facilitate the identification of requirements relevant to a new project. Finally, we recommend the development of a standard taxonomy for transformers, as in the Literature the concepts of *pre-training* and *further pre-trained* often overlap or are misused.

VII. CONCLUSION

In this paper, we proposed SpaceTransformers a new family of three models: SpaceBERT, SpaceRoBERTa and SpaceSciBERT, providing contextualised word embedding for space systems. Our domain specific models were further pre-trained from BERT-Base, RoBERTa-Base and SciBERT-SciVocab on our domain-specific corpus. The pre-trained and further pre-trained models were evaluated on a CR task with our new labelled dataset of space systems concepts. All further pre-trained models outperformed their respective baseline models. The model further pre-trained from RoBERTa-Base, SpaceRoBERTa, considerably improved the F1 score of several labels with a weighted average of 8% with respect to its baseline. The SpaceSciBERT model, further pre-trained from SciBERT-SciVocab, achieved the highest improvement, on the single label, with respect to BERT-Base with an F1 score increase of 12% for the *Communication* label. Finally, SpaceRoBERTa achieved the highest ranking in the Nemenyi CD diagram. The statistical analysis however showed a lack of significant difference in mean ranking for the remaining models.

ACKNOWLEDGMENT

Results were obtained using the ARCHIE-WeSt High Performance Computer (www.archie-west.ac.uk) based at the University of Strathclyde. The authors would like to warmly thank Sabrina Mirtcheva (ESA) and Serge Valera (ESA) for providing the ECSS requirements corpus used for fine-tuning.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, 2017, pp. 5998–6008. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1%c4a845aa-Paper.pdf

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186.

[3] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[4] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 328–339.

[5] A. Aghajanyan, S. Gupta, and L. Zettlemoyer, "Intrinsic dimensionality explains the effectiveness of language model fine-tuning," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process. (Long Papers)*, vol. 1, 2021, pp. 7319–7328.

[6] A. Berquand, F. Murdaca, A. Riccardi, T. Soares, S. Genere, N. Brauer, and K. Kumar, "Artificial intelligence for the early design phases of space missions," in *Proc. IEEE Aerosp. Conf.*, Mar. 2019, pp. 1–20.

[7] A. Berquand, Y. Moshfeghi, and A. Riccardi, "Space mission design ontology: Extraction of domain-specific entities and concepts similarity analysis," in *Proc. AIAA Scitech Forum*, Jan. 2020, p. 2253.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*. [Online]. Available: http://arxiv.org/abs/1907.11692

[9] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3615–3620. [Online]. Available: https://www.aclweb.org/anthology/D19-1371

[10] J.-S. Lee and J. Hsiang, "Patent classification by fine-tuning BERT language model," *World Pat. Inf.*, vol. 61, Jun. 2020, Art. no. 101965. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0172219019300742

[11] J. Krishnan, P. Coronado, H. Purohit, and H. Rangwala, "Common-knowledge concept recognition for SEVA," in *Proc. AAAI Spring Symp. Combining Mach. Learn. Knowl. Eng. Pract. (AAAI-MAKE)*, Stanford Univ., CA, USA, 2020, pp. 1–4.

[12] S. R. Hirshorn, *NASA System Engineering Handbook*, 2nd ed. Washington, DC, USA: NASA, 2016. [Online]. Available: https://www.nasa.gov/sites/default/files/atoms/files/nasa_systems_engin%eering_handbook_0.pdf

[13] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Bioinf.*, vol. 36, no. 4, pp. 1234–1240, Feb. 2020.

[14] C.-N. Chau, T.-S. Nguyen, and L.-M. Nguyen, "VNLawBERT: A Vietnamese legal answer selection approach using BERT language model," in *Proc. 7th NAFOSTED Conf. Inf. Comput. Sci. (NICS)*, Nov. 2020, pp. 298–301.

[15] E. Alsentzer, J. R. Murphy, W. Boag, W. H. Weng, D. Jin, T. Naumann, and M. B. McDermott, "Publicly available clinical BERT embeddings," in *Proc. ofthe 2nd Clin. Natural Lang. Process. Workshop*. Minneapolis, MN, USA: Association for Computational Linguistics, 2019, pp. 72–78.

[16] K. Huang, J. Altosaar, and R. Ranganath, "ClinicalBert: Modeling clinical notes and predicting hospital readmission," in *Proc. CHIL ACM Conf. Health, Inference, Learn., Workshop Track.*, Toronto, ON, Canada: Association for Computing Machinery, 2020, p. 9.

[17] Z. Liu, D. Huang, K. Huang, Z. Li, and J. Zhao, "FinBERT: A pre-trained financial language representation model for financial text mining," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 4513–4519.

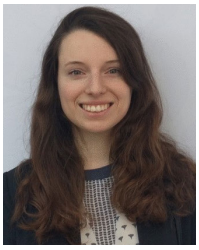
[18] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," in *Proc. Findings Assoc. Comput. Linguistics, (EMNLP)*, 2020, pp. 2898–2904. [Online]. Available: https://www.aclweb.org/anthology/2020.findings-emnlp.261

[19] Y. Si, J. Wang, H. Xu, and K. Roberts, "Enhancing clinical concept extraction with contextual embeddings," *J. Amer. Med. Inform. Assoc.*, vol. 26, no. 11, pp. 1297–1304, Nov. 2019.

[20] Ö. Uzuner, B. R. South, S. Shen, and S. L. DuVall, "2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text," *J. Amer. Med. Informat. Assoc.*, vol. 18, no. 5, pp. 552–556, Jun. 2011.

[21] B. Alex, C. Grover, R. Tobin, C. Sudlow, G. Mair, and W. Whiteley, "Text mining brain imaging reports," *J. Biomed. Semantics*, vol. 10, no. S1, pp. 1–11, Nov. 2019, doi: 10.1186/s13326-019-0211-7.

[22] E. Tseytlin, K. Mitchell, E. Legowski, J. Corrigan, G. Chavan, and R. S. Jacobson, "NOBLE—Flexible concept recognition for large-scale biomedical natural language processing," *BMC Bioinf.*, vol. 17, no. 1, pp. 1–15, Dec. 2016, doi: 10.1186/s12859-015-0871-y.
[23] A. Brack, J. D'Souza, A. Hoppe, S. Auer, and R. Ewerth, "Domain-independent extraction of scientific concepts from research articles," in *Advances in Information Retrieval* (Lecture Notes in Computer Science), vol. 12035, 2020, pp. 251–266.
[24] A. Grivas, B. Alex, C. Grover, R. Tobin, and W. Whiteley, "Not a cute stroke: Analysis of rule- and neural network-based information extraction systems for brain radiology reports," in *Proc. 11th Int. Workshop Health Text Mining Inf. Anal.*, 2020, pp. 24–37. [Online]. Available: https://www.aclweb.org/anthology/2020.louhi-1.4
[25] M. Hofer, A. Kormilitzin, P. Goldberg, and A. Nevado-Holgado, "Few-shot learning for named entity recognition in medical text," 2018, *arXiv:1811.05468*. [Online]. Available: http://arxiv.org/abs/1811.05468
[26] M. Al-Smadi, S. Al-Zboon, Y. Jararweh, and P. Juola, "Transfer learning for Arabic named entity recognition with deep neural networks," *IEEE Access*, vol. 8, pp. 37736–37745, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8993806/
[27] W. Kriedte, "ECSS—A single set of European space standards," in *Spacecraft Structures, Materials and Mechanical Testing*, W. Burke, Ed. Noordwijk, The, Netherlands: ESA, 1996, pp. 321–327.
[28] ECSS Secretariat, "ECSS-P-00C—Standardization objectives, policies, and organization," ECSS Secretariat, Noordwijk, The, Netherlands, Tech. Rep. 1, 2013.
[29] ECSS Secretariat, "ECSS-S-ST-00C: Description, implementation and general requirements," ECSS Secretariat, Noordwijk, The, Netherlands, Tech. Rep. 1, 2020.
[30] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer, and R. Louf, "Transformers : State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6
[31] Technology Coordination & Planning Office, "ESA Technology Tree version 4.0 (ESA STM-277)," ESA Technol. Coordination & Planning Office, Paris, France, Tech. Rep. STM-277, 2020. [Online]. Available: https://esamultimedia.esa.int/multimedia/publications/STM-277/STM-277.pdf
[32] I. Alonso Gómez, "ESA generic product tree (TEC-TP/0045)," ESA, Noordwijk, The Netherlands, Tech. Rep. TEC-TP/0045, 2011. [Online]. Available: http://emits.sso.esa.int/emits-doc/e_support/ESA_Generic_Product_Tree_J%une_2011.pdf
[33] G. Vrbancic and V. Podgorelec, "Transfer learning with adaptive fine-tuning," *IEEE Access*, vol. 8, pp. 196197–196211, 2020.
[34] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith, "Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping," 2020, *arXiv:2002.06305*. [Online]. Available: http://arxiv.org/abs/2002.06305
[35] M. Mosbach, M. Andriushchenko, and D. Klakow, "On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines," 2020, *arXiv:2006.04884*. [Online]. Available: http://arxiv.org/abs/2006.04884
[36] W. A. D. Santos, J. R. Bezerra, L. F. W. Goes, and F. M. F. Ferreira, "Creative culinary recipe generation based on statistical language models," *IEEE Access*, vol. 8, pp. 146263–146283, 2020.
[37] Z. Liu, D. Huang, and K. Huang, "Pretraining financial text encoder enhanced by lifelong learning," *IEEE Access*, vol. 8, pp. 184036–184044, 2020.



AUDREY BERQUAND received the Dipl.-Ing. degree from EPF, France, and the M.Sc. degree in aerospace engineering from KTH, Sweden. She is currently pursuing the Ph.D. degree with the Intelligent Computational Engineering (ICE) Laboratory, University of Strathclyde. Her Ph.D. is half-funded by the European Space Agency (ESA) and in cooperation with AIRBUS, RHEA, and satsearch in the frame of a Networking Partnering Initiative (NPI). She is an Alumnus of the International Space University Space Studies Program. Her research interests include knowledge management and reuse, text mining, natural language processing, and autonomous reasoning for space systems.

PAUL DARM received the Dipl.-Ing. degree in aerospace engineering from Dresden University of Technology, Germany. He recently done his master's thesis about a Knowledge Graph (KG) for space system requirements. He is currently a Research Assistant with the Intelligent Computational Engineering (ICE) Laboratory, University of Strathclyde, working on various applications of natural language processing for space systems.

ANNALISA RICCARDI is currently a Lecturer in computational intelligence with the Department of Mechanical and Aerospace, University of Strathclyde. She has more than ten years of experience in optimization techniques, and machine learning and applications. She is involved in projects on text mining and data-driven decision making for engineering design.