Design Science

Challenges in applying large language models to requirements engineering tasks

Johannes J. Norheim[1], Eric Rebentisch[2], Dekai Xiao[3], Lorenz Draeger[3], Alain Kerbrat[4] and Olivier L. de Weck[1]

[1]*Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA*
[2]*Sociotechnical Systems Research Center, Massachusetts Institute of Technology, Cambridge, MA, USA*
[3]*Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Aachen, Germany*
[4]*Airbus, Toulouse, France*

Abstract

Growth in the complexity of advanced systems is mirrored by a growth in the number of engineering requirements and related upstream and downstream tasks. These requirements are typically expressed in natural language and require human expertise to manage. Natural language processing (NLP) technology has long been seen as promising to increase requirements engineering (RE) productivity but has yet to demonstrate substantive benefits. The recent addition of large language models (LLMs) to the NLP toolbox is now generating renewed enthusiasm in the hope that it will overcome past shortcomings. This article scrutinizes this claim by reviewing the application of LLMs for engineering requirements tasks. We survey the success of applying LLMs and the scale to which they have been used. We also identify groups of challenges shared across different engineering requirement tasks. These challenges show how this technology has been applied to RE tasks that need reassessment. We finalize by drawing a parallel to other engineering fields with similar challenges and how they have been overcome in the past – and suggest these as future directions to be investigated.

Keywords: Requirements engineering (RE); Systems engineering; Large language models; Natural language processing (NLP); NLP4RE1

Received 01 August 2023
Revised 15 March 2024
Accepted 19 March 2024

Corresponding author
Johannes J. Norheim
norheim@mit.edu

Des. Sci., vol. 10, e16
journals.cambridge.org/dsj
DOI: 10.1017/dsj.2024.8

1. Introduction

The shift to ever more complex and coupled engineered cyber-physical systems comes with an increase in the volume of the requirements associated with the system (Norheim et al. 2022). Complex engineered systems such as commercial aircraft may have tens or even hundreds of thousands of requirements representing a wide range of requirements types. As the functionality and complexity of these systems increase over time, especially as additional needs concerning stakeholders, society, or sustainability are added, the burden associated with requirements engineering (RE) will grow correspondingly. Requirements are most often encoded in natural language (NL) – the *lingua franca* of requirements (Zhao et al. 2021).

This is particularly the case with legacy requirements for long-lived systems. As a result, increasing the number of requirements to be managed has often meant a growing demand for human experts to manage them. It has been well established that many of the challenges in the later stages of the product or system development cycle can be traced back to requirements-related problems (Aurum & Wohlin 2005). Although this claim has been mainly studied for software engineering, it also holds for hardware systems (Laplante & Kassab 2022, INCOSE 2015). The most commonly cited challenges relate to properties of requirements: correctness, consistency, and completeness (Rajan & Wahl 2013, Diamantopoulos et al. 2017). However, challenges also stem from eliciting and generating requirements and the downstream tasks that might lead to misinterpretation, oversight, or loss of traceability between design artifacts and requirements. To address the growing volume of requirements and their corresponding burden on engineering resources, natural language processing (NLP) has been considered a potential tool to help with RE tasks. Although NLP techniques long lagged in their capabilities to carry out tasks of even minor complexity, recent developments in the field have led to a renewed interest and hope in the ability of NLP to tackle the many challenges that previously would have been received with skepticism (Berry et al. 2012, Dalpiaz et al. 2018). Although some of the recent renewed interest can be attributed to easier access to general-purpose NLP software, like the NL Toolkit – an openly and freely distributed available repository of Python-based libraries, much comes from a different direction. Significant advances in domain-independent methods in NLP and machine learning (ML) since the inception of transformer-based language models (also known as large language models, or LLMs) like Bidirectional Encoder Representations from Transformers (BERT; Devlin et al. 2018) and the multidomain capabilities of these new tools have translated to adoption in domains such as BioMedicine and Law. As a result, a recent stream of RE research has focused on applying LLMs to various RE tasks. Comprehensive literature review efforts have been conducted on using NLP in RE by Zhao et al. (2021) and even more recently by Sonbol et al. (2022). However, none of these have explicitly focused on LLM-based efforts. Although this literature constitutes a subset of the reviews, LLMs come with inherent challenges in their adaptation to RE, and we believe this point has received too little attention so far. One notable exception is the work of Deshpande et al. (2021), which focuses on one particular challenge of applying LLMs. In this article, we seek to understand better the broader inherent challenges of applying LLMs in the RE field. Some of these are specific to the technology of LLMs, while others are inherent to the field of NLP.

The article is structured as follows. Section 2 gives additional background on RE as a standalone process, applying NLP to requirements, and the newfound hopes of LLMs. Section 3 summarizes studies of using NLP for RE and explores how the approaches correspond to typical RE tasks. We address the approaches of these studies: the scale of the requirements, the dataset to which the NLP methods were applied, the validation of studies, and how much the task reflects a real-world task required by practitioners, such as RE. We introduce five categories of RE tasks and investigates potential LLM applications within each task. Section 4 summarizes the challenges identified in the studies from Section 3. Section 5 discusses the insights gained from this review of the prior studies and potential gaps between published research and the requirements of engineering tasks. It gives recommendations for future research to overcome these challenges, inspired by fields where the

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

application of LLMs has matured. In the conclusion, we summarize our findings, provide limitations and highlight additional opportunities for future research.

2. Background

Writing requirements constitutes an essential part of the engineering process: "Effective requirements engineering lies at the heart of an organization's ability […] to keep pace with the rising tide of complexity" (Hull et al. 2005). They are captured at different stages of the maturation of the design, from stakeholder analysis and high-level system requirements down to detailed component specifications (Hirshorn et al. 2017). This encompassing/central role of requirements has made them the focus of RE processes that detail how to generate requirements, how to manage requirements once they have been created, and how to link artifacts resulting from a large set of downstream tasks, including design, validation, and verification, back to the requirements.

Requirements can be encoded through textual or graphical supports (Bruel et al. 2021). Textual requirements lie on a spectrum from informal to semi-formal to formal. Informal requirements are written in NL with minor constraints (e.g., using certain modal verbs like shall). Semi-formal requirements are typically structured through boilerplates and templates (Mavin et al. 2009, Dick & Llorens 2012, Rupp 2014, Hall et al. 2020), or patterns (Rajan & Wahl 2013). Formal requirements like temporal logics are specified by a formal context-free grammar with mathematical semantics. Graphical requirements are typically defined through different types of diagrams that also range in formality, from sketches capturing the intention of the requirements to visual modeling languages like universal modeling language (UML) and systems modeling language (SysML).

Because of the ubiquity of informal and semi-formal natural language-based requirements, NLP has been viewed as a fitting technology for many engineering processes that link to requirements. As early as the 1990s, Ryan (1993) proposed using NLP for RE (for information systems) but with a limited scope of application from the perspective of the time. The two applications envisioned were related to scanning support documents to assist with the requirements definition process and traceability maintenance "to guard against their [the requirements] being lost." Since then, a more extensive set of applications have been proposed (Kof 2005) with a fuzzy distinction between applications driven by domain-independent NLP use cases versus challenges trickled down from the RE domain.

Typical domain-independent NLP applications have included sentence classification (a common use case being sentiment classification – for example, of user reviews and comments), question-answering, named entity recognition (NER), text summary, and next sentence prediction (for example, for applications such as chatbots). In a RE context, some of these applications, notably text classification and applications that resemble NER and that might go under different names, such as concept recognition (Berquand et al. 2021), have primarily been driven by the inspiration of the function and perceived potential of these more general NLP techniques. On the other hand, techniques such as concept extraction and traceability detection have typically been driven by a need from RE to potentially benefit from using existing NLP technologies.

In addition to a large body of research on NLP of engineering requirements, these past efforts have recently been surveyed by Zhao et al. (2021) and by Sonbol et

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

al. (2022). In their work, they created a classification of NLP applications. They noted recent trends, particularly the steady increase of research attention directed toward NLP applied to RE tasks, and coined the acronym NLP4RE. Beyond classifying the applications by RE task, these surveys also classify the applications based on the NLP technique employed. These methods can broadly be divided into rule-based (predominant at the inception of NLP) or machine-learning-based methods.

Transformer-based LLMs dominate the current state-of-the-art in NLP (Manning 2022) and belong to the machine-learning-based class of methods. As the name suggests, LLMs can achieve such results based on the massive amount of data they were trained on, albeit coupled with high computational costs. The evolution of these LLMs has been rapid. The concept of transformer-based LLMs was initially proposed by Vaswani et al. (2017) and materialized with the first widely adopted pre-trained LLM: the BERT language representation model introduced in 2018 by Google (Devlin et al. 2018). OpenAI released the pre-trained LLM generative pre-trained transformer (GPT) in the same year (Radford et al. 2018). OpenAI's GPT-2 followed in 2019, GPT-3 in 2020, GPT-3.5 in 2022 and GPT-4 in 2023. Each of these new LLMs were released based on an order of magnitude (or greater) increase in the number of parameters in the model used to perform linguistic tasks with increasing accuracy, recall, and capability. The rapid increase in LLMs' capabilities has stoked interest in their application to tasks such as RE. However, their rapid and recent development has meant that research and publications have struggled to keep up and are relatively few to date.

A significant advantage of LLMs is that once created, they can be tailored to specific applications, allowing reuse of their general capabilities and avoiding the considerable expense of recreating them. To allow the reuse of these models for particular applications, the processes of pre-training and fine-tuning for language models were introduced (Howard & Ruder 2018). It is helpful to split these tasks into four steps: general pre-training, domain-adaptive pretraining (DAPT), task-adaptive pre-training (TAPT), and fine-tuning. In the pre-training step, a model is trained on a general-domain corpus dataset (e.g., BERT was trained on the order of 10^9 words) to capture the general features of the language in the different model layers. While this stage is the most resource-intensive, it must only be performed once. In the following step, DAPT, the model is adapted to the desired domain using domain-specific data (e.g., SciBERT, a modified version of BERT further trained on a corpus of scientific publications with 10^8 words). In the TAPT step, the same process is carried out with task-specific data (e.g., ReqSciBERT based on about 10^4 words from a corpus of system requirements statements (Lim 2022)). None of the steps so far requires any labeled data. Subsequently, transfer learning is applied in the fine-tuning step, where the pre-trained model is fine-tuned on labeled task data (e.g., 10^3 words from about 100 sentences from a labeled requirements statements dataset), hence capturing the specific details of the task (Gururangan et al. 2020). Fine-tuning requires a human expert (preferably multiple human experts to reduce bias) to label a dataset and supervise the training. Throughout the steps from general pre-training to fine-tuning, the required training data decreases steadily, and so does the computational cost. Thus, with each further step, the overall effort needed for adapting the LLM to new tasks (primarily computational resources) decreases, while the human-based proportion of that effort generally increases.

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

The tradeoff for this increasing level of effort is that the trained and tuned model exhibits higher performance than a general model measured by NLP metrics, such as the F1 score. The F1 score computes the unweighted average of the model's precision and recall at specific NLP tasks. Precision measures the degree to which a model overpredicts a data class (based on true and false positives) as a proportion of the total predictions. Recall measures the degree to which a model underpredicts a data class (based on true positives and false negatives) as a proportion of the total predictions. A model with perfect precision and recall would have an F1 score of 1.0. This score is often used to measure the effectiveness of models developed for specific purposes, including examples of tailoring LLMs for RE tasks.

3. Requirements engineering tasks and large language models

This section reviews existing studies to identify candidate RE tasks that may benefit from using NLP and LLMs. Note that these categories do not necessarily reflect specific use cases for NLP methods requested by RE practitioners but rather reflect the tasks that have been explored and may be of potential benefit. This summary of RE tasks is based primarily on the comprehensive reviews by Zhao et al. (2021) and Sonbol et al. (2022). Each review identified different RE task categories based on their survey of the literature: Zhao et al. (2021) divided RE tasks into six categories, four derived from Berry et al. (2012): detection, modeling, tracing and relating, extraction, and two additional ones: classification, and search and retrieval. Sonbol et al. (2022) derived five categories instead: analysis, quality, extraction, modeling and management.

This article proposes a modified classification illustrated in Figure 1 and summarized in Table 1. The categories are based on a synthesis of the categories identified by Zhao et al. (2021) and Sonbol et al. (2022), while framing them in the context of the RE process based on Hickey and Davis (2004) and van Lamsweerde (2009). The connection between the artifacts and the different tasks is explicitly given in Figure 1. The task description is aligned with the generative perspective of LLMs: detailing text is given as an input and what text is generated as an output.

3.1. Generation

In the context of LLMs, *generating requirements* is the task that given NL text as an input results in proposed requirements for the system to be designed. We broaden this category to include uses of LLMs that help formulate the requirements and assist in elicitation. Following the model from Hickey and Davis (2004), this includes any form of LLM that assists in generating situational information, including characteristics of the problem, solution, and project domain, that results in the generation, either by a stakeholder, engineer or LLM of additional requirements. The input information to the LLM could consist of stakeholder needs captured in textual form, user reviews, and requirements and specifications from past or current projects when the goal of reusability is targeted. In this category, we also include refining past or existing requirements into textual forms that would result in new requirements. This includes generating requirement templates or core requirements that might be shared across product lines with variability in the

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

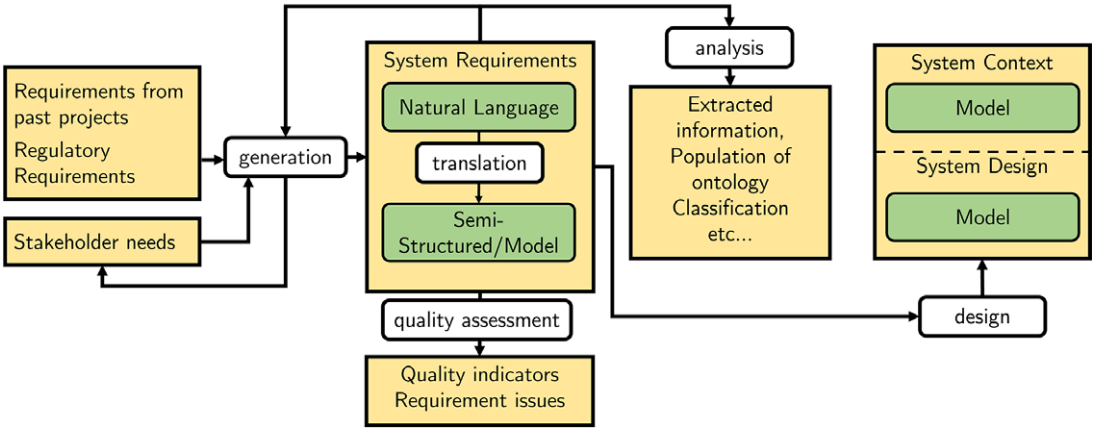

Figure 1. Requirements engineering tasks and their connection to key inputs and outputs from a model-based systems engineering (MBSE) point of view.

Table 1. Requirements engineering tasks classification from a large language model perspective

RE task	Task description
Generation	Generate requirements, either in natural language or any structured form (including formal requirements), based on input text from which a requirement can be derived or extracted.
Translation	Take a requirement and produce an equivalent form, e.g., in a formal language, a semi–structured form (i.e., conforming to boilerplates, templates, and patterns), or a requirements modeling language.
Quality assessment	Find common quality problems with requirements.
Analysis	Derive non–requirement information useful for downstream engineering tasks based on input requirements.
Design	Propose designs that satisfy one or multiple requirements.

sense discussed by Reinhartz-Berger and Kemelman (2020). In this sense, *generation* is meant to be more general than the *extraction* category proposed by Sonbol et al. (2022) and overlaps with the *search & retrieval* category by Zhao et al. (2021). According to Sonbol et al. (2022), one-sixth (16%) of existing NLP RE research covered this use case. Still, the applications of LLMs, more specifically, remain limited. For instance, de Araújo and Marcacini (2021) aimed to automate the identification of software requirements based on user reviews. This effort developed RE-BERT (Requirements Engineering using Bidirectional Encoder Representations from Transformers) by fine-tuning BERT with a token classification model that classified every part of a sentence in a review as referring to an existing software product functionality or not. Reviews referring to existing functionality could then lead to future performance requirements. The data set was generated based on online user reviews of eight mobile phone applications and included 1000 requirements. LLMs open the unexplored possibility of elicitation through the

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

fine-tuning of assistants (Rastogi et al. 2020) that employ interactive chats to guide a stakeholder to discover new requirements. This could leverage existing elicitation techniques, for example, user stories (Lucassen et al. 2016), or many other techniques (Hickey & Davis 2004). Recent research has been carried out in this direction: Arora et al. (2023) used a pre-trained LLM based on GPT-4 without further tuning and through zero-shot prompt strategies alone.

3.2. Quality assessment

This category maps to the *detection* task from Zhao et al. and the *quality* task from Sonbol et al. *Quality assessment* can be addressed at the individual requirement level: ambiguity and conformance to requirement writing guidelines or syntactic structures defined by boilerplates, templates, and patterns. Quality assessment can also be addressed at the requirement document level, aiming to find missing requirements (addressing incompleteness) and finding redundant, unnecessary, or conflicting requirements (addressing consistency). According to Sonbol et al. (2022), one-sixth of existing NLP RE research covers finding quality issues in requirements. Recent methods (Draeger 2023, Malik et al. 2022) have proposed the application of LLMs' inherent capability for detecting semantic similarity between words and sentences to find similar requirements that might either point toward a conflict or a redundancy. Luitel et al. (2023) proposed using LLM unsupervised training mechanisms to address the completeness of individual requirements. They carried out the training based on masked language model techniques (Devlin et al. 2018), which consist of training the language model to predict missing words (filling in the blank) based on context. For training and testing, they used a subset of the PURE dataset, and since the method was unsupervised, no annotations were created.

3.3. Analysis

Requirement analysis addresses any derived information that could be used in aiding with a task downstream of the creation of the requirement or in connecting a new requirement to already existing information. This includes analyses specific to NLP methods, such as NER, which is not specifically an RE task but can produce information that can be used in RE analyses. This definition is similar to that given by Sonbol et al. (2022) and includes the tasks of traceability – which is important in the context of validation and verification tasks and requirement classification. We differ from Sonbol et al. (2022) by including in this category the task of information extraction (IE), which includes semantic role labeling tasks like NER. This category overlaps with the modeling category from Zhao et al. (2021) and Sonbol et al. (2022) for derived modeling information (i.e., generating Use Case models in UML/SysML). Analysis is the task area that has received the most attention from LLMs so far. Most applications have focused on requirement classification, assigning requirements to a predefined set of classes conventionally defined in RE. Traceability, the task of finding connections between a requirement and other requirements or design artifacts, is also an application that has attracted LLM research, especially due to the key element of LLMs encoding semantic information. In addition, we are

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

also starting to see more information extraction applications like NER and relation extraction.

The most successful LLM application to RE so far has been classification tasks. Hey et al. (2020) developed NoRBERT, a BERT model fine-tuned with a sequence classification model based on the PROMISE dataset. They showed that LLMs could outperform state-of-the-art methods results, like those from Kurtanović and Maalej (2017). Kici et al. (2021) also fine-tuned a sequence classification model based on eight different categories. The information extraction task has mainly focused on NER to extract concepts belonging to abstract categories relating to language construction (e.g., objects, functions, modifiers). This can be formulated as a token classification task for LLMs, which requires only minor modifications to the original BERT neural network architecture. Berquand et al. (2021) adapted BERT to generate SpaceBERT, a domain-adaptive version of BERT for space systems engineering. They then fine-tuned it to extract space systems concepts from 18 categories from a requirement corpus. Ajagbe and Zhao (2022) carried out multi-label classification for nine concept categories: object, agent, container, instrument, conditional, temporal, location, goal and dative. Tikayat Ray et al. (2023) developed aero-BERT-Classifier, and aero-BERT-NER, together referred to as aero-BERT. These two models are fine-tuned versions of BERT on requirement classification and NER. The NER task was carried out for five categories, and the aero-BERT-Classifier was trained to assign six requirement categories. Chami et al. (2019) carried out NER for six entity categories derived from SysML. Bajaj et al. (2022) extracted UML cases based on requirements using GPT-3.

Relations between requirements and other requirements, or derived artifacts (design models, test procedures, etc.), have been an active area of research in RE. Recently, LLM has been applied to this RE task as well. Deshpande et al. (2021) developed RDC-BERT, a fine-tuned BERT model, on a binary classification task to detect whether two requirements were linked. They show that the BERT model outperforms alternative ML models like decision trees. However, the article argues that the return on investment of the more expensive training and data-hungry requirements for BERT only outperforms more conventional models after a particular scale of the annotated dataset is achieved. Fischbach et al. (2021) developed a simpler binary classification method to detect whether requirements contained causal sentences. Lin et al. (2021) mapped traceability between code issues (which could be interpreted as requirements) and implemented code.

3.4. Translation

Requirement translation is motivated by the underlying interpretation of NL translation in the context of LLMs: a mapping from one set of words (or tokens) to another set of words (or tokens). There might be some overlap between translation as defined here and the modeling category proposed by Zhao et al. (2021) and Sonbol et al. (2022). Translation is most commonly applied to translating requirements from one NL to another. It can also be applied to translation from natural to artificial languages, such as a requirement modeling language. In the extreme, the input to be translated might even be an informally formulated requirement that is transformed into a derived requirement in a formal or semi-formal language. Although existing NLP applications to RE tasks have not yet been

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

categorized as translation, some of the non-LLM applications that have been categorized as modeling would fall into this category, for example, the applications explored by Deeptimahanti & Sanyal (2011) and Dawood & Sahraoui (2017). The number of applications of LLM to this category of RE tasks remains small. One area of application that has received recent attention is the translation of NL Linear Temporal Logic (LTL) specifications to their formal counterpart. The NL requirements could be expressed in unstructured (Cosler et al. 2023) and structured (He et al. 2022) forms. Methods applying LLMs range in methodologies. (Hahn et al. 2022) fine-tuned an LLM (T5, an encoder-decoder transformer) to translate NL specifications of RegEx, First Order Logic, and LTL statements into their formal counterpart. Cosler et al. (2023) used a few-shot approach with a GPT-3 variant, with a step-by-step instruct approach. They tested the performance with 36 edge-case specifications generated by five experts, achieving a best-case of correctly translating 31 specifications.

3.5. Design

We refer to *design* as the task that outputs design artifacts, given validated requirements as an input. This is a new category that traditionally would not be included in NLP applications, but we include it as we expect this to become a growing application enabled by LLMs. Although the derivation of design that satisfies requirements would often be conceived as an exclusively human task, with the advent of generative AI, we can now start envisioning LLMs deriving textual descriptions of the design, either in NL or formal design modeling languages in the context of MBSE. We did not find any published work that specifically targets the use of LLMs to generate design artifacts based on a *set of requirements alone*. This category overlaps with the *general usage* of LLMs to generate design artifacts: this includes methods that assist in generating new concepts, exploring the design space, or proposing alternative designs based on non-requirement information: maybe preferences or domain knowledge of engineers. LLMs are already being applied in this domain: Zhu and Luo (2022) explored the usage of the GPT LLM to generate novel design concepts based on high-level textual problem descriptions. Code synthesis, a now widely adopted technology, outputs valid code in various programming languages based on a short description of what the code should accomplish Austin et al. (2021). We exclude this broader treatment of the design category, as it is beyond the scope of this article, and reserve it exclusively for cases with correctly formulated and validated requirements as the sole input. Although the literature in this domain, when focusing on requirements, is still scarce, we predict significant growth.

3.6. RE task-independent work

Although not an RE task, the fine-tuning applications discussed in the previous RE task categories rely on pre-trained LLMs. Although general-purpose LLMs such as BERT or RoBERTa can be used out of the box, fine-tuned models can perform better on domain-adapted or task-specific adapted pre-training, which is a task that can be carried out independently. For this purpose, Ajagbe and Zhao (2022) trained BERT4RE, a domain-adapted version of BERT for RE, on the PURE and PROMISE datasets and online review. Berquand et al. (2021) trained

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

Table 2. Overview of specific LLM applications

Publication	Category	Task description	Relevance for downstream task
de Araújo and Marcacini (2021)	Generation	Finding references in online user reviews to existing requirements	May be used to elicit further requirements.
Hey et al. (2020)	Analysis	Classification of requirements into functional and non–functional classes.	Not explicitly stated
Kici et al. (2021)	Analysis	Classification of requirements into eight custom–defined classes.	Not explicitly stated
Dalpiaz et al. (2019)	Analysis	Detection of the presence of quality or functional aspects.	Not explicitly stated
Berquand et al. (2021)	Analysis	Extraction of concepts belonging to 18 categories.	Domain modeling
Chami et al. (2019)	Analysis	Extraction of concepts belonging to six categories.	Identifying elements of the context/design model
Ray et al.2023)	Analysis	Classification into design, functional or performance categories, and extraction of concepts belonging to five categories.	Detection of missing information given expected information for standard categories
Fischbach et al. (2021)	Analysis	Dependency detection between requirements.	Assist in redundancy and inconsistency detection
Lin et al. (2021)	Analysis	Detecting trace from requirements to design artifacts.	Complying with traceability standards
Cosler et al. (2023)	Translation	Translating natural requirements into temporal logic.	Model checking (simulation–based verification)

SpaceBERT and several variants (e.g. SpaceRoBERTa), a domain-adapted version of BERT trained on documents from the space domain, including requirements documents.

3.7. Summary

We summarize the findings from these five subsections in Tables 2 and 3. Table 2 focuses on the actual applications, while Table 3 focuses on details with respect to the data and LLM used.

R stands for reused (from previous study or publicly available annotated dataset), A for automatic (when the annotations can be generated based on simple rules from existing data), and M for manual annotation. LLM Architecture legend: B stands for binary classification, C for classification with more than two classes, TC for token (entity) classification. P stands for prompt engineering.

Some common themes emerge from this review of applications of LLMs to RE tasks. First, the number of requirements analyzed in the studies ranged from a few dozen to almost 2000. These numbers are relatively small in the context of the size of requirements sets typically found in complex systems development efforts.

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

Table 3. Overview of LLM publication with the number of requirements, training, validation and testing split, annotation method, and architecture

Publication	Category	No. of Requests	Train/valid/test	Annotation	LLM architecture
de Araújo and Marcacini (2021)	Generation	1000	87.5/12.5/0%	M	BERT + TC
Hey et al. (2020)	Analysis	625	469/156/–	R	BERT + C
Kici et al. (2021)	Analysis	Not specified	80/10/10%	–	DistilBERT + C
Dalpiaz et al. (2019)	Analysis	1502	469/156/877	R + M	BERT + C
Berquand et al. (2021)	Analysis	882	80/20/–	M	BERT + TC
Chami et al. (2019)	Analysis	100	80/–/20%	M	BERT + C
Ray et al. (2023)	Analysis	325	90/10/0%	M	BERT + C and BERT + TC
Fischbach et al. (2021)	Analysis	61	Not specified	M	BERT + B
Lin et al. (2021)	Analysis	1834	Not specified	A	BERT + B
Cosler et al. (2023)	Translation	36	Not specified	M	GPT3 variant + P

However, finding open-source requirements datasets from large-scale systems can be very challenging, and the search is often limited to academic sources (e.g., engineering competition projects) or system-level requirements for publicly-funded programs (e.g., infrastructure or science). Another theme is that of the five RE tasks defined here, the Analysis task is by far the most commonly addressed in these studies. Of the studies in this task grouping, most explore NLP-specific operations such as sequence classification or NER rather than addressing RE tasks directly. This is consistent with the task's definition, where analysis produces products that potentially aid in downstream RE or engineering efforts but explores NLP technology feasibility ahead of the application to and refinement of RE tasks in practice.

One objective of this study was to identify RE tasks that could benefit from LLMs. To date, we are unable to cite studies that directly explore the application of LLMs to these RE tasks. New LLM technologies and particularly generative language models like the recent variant of GPT (GPT-4) are very new, and systematic studies of their potential to help RE are struggling to keep up with the rapid pace of their evolution. Nevertheless, given the broad interest in LLMs in general and the exploration of NLP in RE, we anticipate that new studies exploring the applicability of LLMs to RE tasks will eventually emerge.

4. Challenges to using large language models in requirements engineering

As of this article's writing, the application of LLMs to RE is still in its infancy. As noted previously, LLM capabilities have advanced rapidly in a short period, and the

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

number of published studies documenting their RE application is small. However, adoption and deployment challenges of NLP have long existed in other domains and the more general field of artificial intelligence. It is perhaps useful to learn from deployment experiences in other domains to better anticipate where challenges might arise when applying LLM-based NLP methods for RE. In the biomedical field, Chapman et al. (2011) identified a set of six challenges: lack of access to shared data, lack of annotated datasets for training and benchmarking, lack of reproducibility, insufficient common conventions and standards for annotations, lack of user-centered development and scalability, and lack of collaboration. To the best of our knowledge, a dedicated study addressing the challenges identified by Chapman et al. for RE is yet to be published. We hypothesize that NLP applications become more attractive to improve RE productivity in the cases of large volumes of requirements generated by advanced systems engineering. We seek to identify possible challenges to implementing NLP methods in RE and use as a starting point the six challenges identified by Chapman et al. To simplify the discussion, those six challenges can be generalized into the categories of challenges with *limited requirements-specific data*, *inconsistent data annotation* and *inadequately defined RE use cases*. These types of challenges will be explained in the sections that follow. The data challenges identified as problematic in other domains may be especially relevant to this discussion, given that the performance and accuracy of large-language-based methods are critically dependent on supervised learning, large volumes of sample training data, and fine-tuning.

4.1. Limited requirements-specific data

Open datasets for requirements remained a long-time challenge until recently. As pointed out by Ferrari et al. (2017): "With some exceptions, most of the works use proprietary or domain-specific documents as benchmarks, and replication of the experiments and generalization of the results have always been an issue." Although publicly available datasets have long existed, a central and diverse repository for the community to work on did not. To address this issue, the PURE dataset was published by Ferrari et al. (2017). This dataset composes a corpus of 79 raw public requirements documents curated from the Web from a wide range of products and systems. One other source of requirements to address this challenge is the PROMISE requirement dataset, comprising 625 requirements and released in 2007 based on 15 student projects (Cleland-Huang et al. 2007). Yet, although diverse in application types, all of the documents come from software engineering. Even though the requirements are often for a hardware system's software component or control, hardware engineering requirements are not represented in these datasets.

Is it helpful to differentiate between requirements for software-based and hardware-based systems? That is an empirical question to be addressed. Indeed, hardware requirements represent a critical, if not dominant, part of the engineering of advanced systems. Hardware requirements are arguably different from software requirements in that software is often developed with a modular architecture while hardware may be driven by performance or other drivers toward more integrated architectures. The coupling associated with a more integrated architecture may create additional dependencies between requirements, perhaps greater use of domain-specific knowledge, system architecture knowledge, or other context-specific information not directly referenced in the requirement statement. Since

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

LLMs are trained and fine-tuned based on available data, any bias in the data sampling (e.g., hardware, software, etc.) may distort the model's outcomes. For instance, is the corpus of requirements statements alone sufficient to train and fine-tune LLMs or do other sources of context- and system-specific data need to be included in the training corpus in order to produce a helpful LLM for RE tasks?

Additional public datasets that have been used comprise public user review (de Araújo & Marcacini 2021), and public standards, for example, Part 23 and Part 25 of Title 14 of the Code of Federal Regulations for aircraft (Tikayat Ray et al. 2023), and the European Cooperation for Space Standardisation (ECSS) dataset. The most extensive coverage, to our knowledge, of recent hardware requirements was carried out by Lim (2022), containing 42 hardware systems, including two communication systems, five telescope systems, and 35 spacecraft systems. Data challenges have primarily been addressed in software and control requirements. Yet a diverse set of publicly available hardware requirements across engineering industries (aerospace, automotive, communications, oil and gas, energy, infrastructure, marine) has yet to be curated. Although these fields all have certification standard documents that include requirements, these are typically behind paywalls and not publicly available.

One aspect of data challenges not typically mentioned in the literature is the diversity in requirements databases, depending on what type of requirement is given and from what part of the system and the system lifecycle it derives. For instance, a system-level requirement (generally a small proportion of the total requirements) may be expressed concisely, with clear functions, targets, measures, and relationships. On the other hand, a derived requirement (perhaps for a subsystem or component and likely representing the majority of requirements in a large requirements database) may be wordy, context-specific (with many expressed or implied dependencies to other elements in the system), and may include a combination of functions, operations, references to other documents, or a variety of different data representations. The latter type of requirement poses a different and more demanding challenge to NLP processes than the former based on the complexity of the expression, and may require a different NLP approach and use case.

4.2. Inconsistent data annotation

Of the publicly available datasets, most of the requirements (i.e., the PURE dataset) are not annotated for requirements engineering-specific tasks. PROMISE is one of the few exceptions – yet large annotated corpora, similar to those used for grammar and syntax NLP tasks, such as Penn Treebank or the CoNLL-2003 NER task (annotating names of persons, locations, organizations, and miscellaneous entities) (Sang & De Meulder 2003), that are used for benchmarking and reproduction of results of general purposes NLP tasks, are too rare.

The lack of annotated datasets has two major drawbacks for the fields of RE: first, the lack of benchmarks, when using the data for testing, and second, the lack of training data for ML purposes.

The lack of benchmarks is critical, as it makes it impossible to compare multiple NLP methods that attempt to solve the same task, and that can thereby claim that they are able to solve this task beyond a set of crafted use cases. The risk is that the method overfits to the specific test case for the domain used in a study, and does not

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

generalize to other domains. The lack of benchmarks also makes it hard to benchmark the novel LLMs against past methods that have been applied (e.g., rule-based, feature-based ML) (Lim 2022).

Second, the lack of annotated datasets used for training data is even more problematic when applying LLMs. Fine-tuning and few-shot approaches (that is with GPT family models) require training data (that is examples) as an input. Whereas rule-based methods rely on a set of handcrafted expert rules that can be validated against a small set of use cases and tests, LLMs require annotated data for fine-tuning. Although new architectures, like GPT, have enabled zero-shot, or few-shot applications, these are still poorly understood and covered in the literature, and the lack of adequate training data requires a significant annotation effort for every study. As a result, although NLP methods in the past could get away with the lack of training data, with LLMs this is (exceptions aside) no longer the case.

One challenge very specific to LLMs is turning RE tasks into LLM problems. This involves either adapting the underlying architecture of the LLM by adding a few neural network layers or, in the case of GPT-based LLMs, the careful task of prompt design and prompt engineering (that is, tailoring the model prompts or instructions so they are more compatible with the structure of the LLM). For LLMs deriving from the encoder transformer architecture, like BERT, the challenge is to frame the NLP task as a sequence classification, sequence labeling, or sequence-to-sequence problem. For example, Soares et al. (2019) framed the task of extracting relationships between entities with BERT in six different ways, where a naive choice versus a more advanced architecture could differ by as much as 46.6 percentage points in the F1 score.

4.3. Inadequately-defined Requirements Engineering use cases

The application of NLP within the RE tasks discussed in Section 3 has too often been driven by what we would describe as technology push rather than a need pull. That is to say, at this early period in the evolution of LLMs, published studies (or unpublished but public presentations or discussions) tend to focus on trying to understand and demonstrate what the LLM capabilities can accomplish when applied to NL requirements. This is the technology push aspect since the technology solution is searching for an application.

The motivation behind specific use cases can often be called into question when we ask what will be done with the data returned from the NLP application. Most of the case studies used in the literature stem from 'what-if' scenarios, where NLP practitioners seek the application of their techniques to RE. Less often are the case studies inspired by either trying to reproduce a repetitive RE task carried out by engineers in the practice already or solving a problem that RE practitioners are constantly faced with.

The lack of clearly defined use cases also makes it difficult to make claims of success on the basis of F1 scores alone, which gives equal weight to precision and recall. As pointed out by Lim (2022), "the failure to extract all required information may result in the construction of an incomplete system model, affecting its functionality and desired ilities. In this specific context, recall should be prioritized over precision." Berry et al. (2012) make a similar claim that for certain applications, nothing short of a perfect recall and perfect precision would be acceptable – the *Perfect Recall Condition* according to Lucassen et al. (2016), as the tool should

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

not miss any critical information, or mislead the user with wrong information. The utility of any NLP, and by extension LLMs, for RE, therefore, is also highly dependent on the downstream task that needs to be carried out based on the result of the output generated.

4.4. Adoption challenges

Adopting LLM-based NLP tools for RE will require new skill sets that may not currently exist among RE professionals, particularly in ML methods and tools, model training and validation, and implementing those methods and tools to address specific use cases. Critically, the current performance of the best LLM-based NLP tools is less than perfect in precision and recall (i.e., the F1 scores of the best models are less than 1.0). That means these models will continue to produce false positive and false negative predictions for the foreseeable future. Organizations will have to determine in which RE applications the error level is tolerable and in which cases it is not. For instance, would the imperfect prediction performance of LLMs be acceptable in highly regulated environments or where human safety is an overriding consideration? With additional fine-tuning and training, the performance of these models will improve, but that will likely require RE experts to be directly involved in the training and validation of the models. Experts are frequently in short supply in most organizations. They may perhaps not be available, allowed, or inclined to commit time to develop the application of these NLP tools unless they are considered to be a priority by the organization.

5. Discussion

NLP and, more specifically, transformer-based LLMs are relatively new technologies advancing rapidly. Progress in the field is being made both in academia and in industry. Notably, private companies are investing heavily in the development of the technology and each new generation of LLM, and in many areas, are leading the state-of-the-art by virtue of their ability to collect and apply the massive amounts of data needed to train LLMs. The allocation of focus, effort, and contributions is not as clearly differentiated between academia and industry in the advance of LLMs as in more mature product domains where industry might play more of a leading role. Therefore, the implications resulting from this review are not necessarily for academia and industry but for NLP for RE practitioners and NLP4RE researchers.

For the RE practitioners interested in the potential of NLP, there is a need for clearly defined RE use cases in which NLP methods can be applied. These use cases should have clearly defined outcomes and measures of performance to enable comparison of the benefits not only with various NLP approaches but also comparison of NLP with traditional RE methods to determine how the most compelling benefits might be obtained. Absent a definition of specific use cases, the tools' need for accuracy, speed, and effectiveness, and the required outcomes for technical and program management and other stakeholders, it will not be easy to judge whether NLP LLM-based techniques provide compelling benefits to RE tasks compared with existing approaches.

The F1 score is a valuable measure of the performance of a language model but a poor indicator of the overall benefit resulting from the application of the model to a real RE task. Much of the existing literature reports analysis based on measures of

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

model performance rather than benefits from application. This is not unexpected given the relative infancy of LLMs applied to RE. Still, studies going forward should identify specific application use cases that enable comparisons between NLP and traditional methods based on outcomes and relative benefits. In addition to the F1 score to assess the relative performance advantages of the model, the performance measures for the RE tasks themselves should be a part of assessing the relative benefits provided by applying NLP methods.

RE practitioners are also in a favorable position to identify and curate requirements datasets representing the range of requirements types encountered in practice. These datasets may be considered confidential and competition-sensitive, but through collaboration with academic researchers or industry consortia, standardized or benchmark requirements datasets could be developed that can enable consistent evaluation of the performance of various NLP approaches or methods across multiple industry or academic research teams.

For NLP4 RE researchers (and NLP researchers in general), there is a need to develop additional measures of model performance beyond the F1 score better to understand the practical impact of these emerging methods. Also, the tendency of ML models to produce biased results based on the sample datasets used in their training has been demonstrated across various application domains, such as image recognition, text analysis, and recommendation engines. There is great diversity in requirements depending on, e.g., their source, type, place in the allocation hierarchy, degree of embeddedness in a specific system or technological context, and so forth. It would be valuable for researchers to explore the degree to which this diversity in the language used to express requirements potentially creates a bias that limits the effectiveness of NLP methods for RE.

5.1. Implications for future work

The challenges discussed in Section 4 are challenges that have existed in other NLP domains (Chapman et al. 2011) but also in ML areas beyond NLP. To help address these potential challenges to the application of LLMs to RE, we recommend that future studies should include:

- Increased use of hardware requirements datasets that include the increased degree of dependency that can come with hardware compared with software requirements.
- Comparing the challenges that requirements of different types and from different stakeholders or different stages of the system lifecycle might pose to NLP and LLM methods, and how to tailor them accordingly. The PURE dataset classifies the requirement datasets into high-level and low-level requirements. The curation of additional requirements datasets that include requirements of multiple types based on the same classification approach would help to understand how diversity in requirements types and language might affect the benefits of NLP and LLM methods in RE tasks.
- Develop benchmark requirements datasets to enable the systematic exploration and evaluation of the application of NLP and LLM methods to different RE tasks to understand which of different approaches to the application of these technologies, if any, provides the greatest benefits. In the NLP field of information extraction (IE), a dedicated set of challenges, sponsored by NAVWAR, the US

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

Navy authority and acquisition command (formerly NRAD), created a common set of annotations for NER for so-called coarse-grained categories of information: organization, person names, locations, time, currency and percentages (Li et al. 2022, Grishman & Sundheim 1996). A larger dataset with similar categories, CoNLL-2003 NER challenge, was based on the same idea. The SemEval (Semantic Evaluation) set of conferences has led to a set of benchmarking cases and annotated dataset that has led to many follow-up studies, for example, SemEval 2010 Task 8 "Multi-Way Classification of Semantic Relations Between Pairs of Nominals".

- Identify RE tasks and reference use cases in the context of a real Product Development Process to design and deploy NLP and LLM methods and evaluate their strengths and limitations relative to existing methods.

6. Conclusion

The volume of requirements and requirement-related data will only increase with the advent of more complex advanced systems. Whereas many existing systems can still rely on the human effort of expert engineers in RE and systems engineering to carry out the tasks manually, this approach will likely not scale without additional tools and methods. NLP has long been envisioned as the solution to this problem, however, has yet to be adapted as a state of the practice for RE tasks.

In this article, we reviewed the recent development of state-of-the-art NLP LLM-based techniques for RE tasks that have gathered significant traction. As noted, while a number of possible RE applications have been explored, many are based on demonstrations of the capabilities of the NLP technologies rather than a systematic evaluation of what RE tasks are suited to the capabilities and limitations of NLP methods. Understanding the net benefit in terms of improved RE and system outcomes when counted against the cost to deploy the NLP methods is underexplored.

The needs for precision and recall, speed, and effectiveness from the tools, and the required outcomes for technical and program management and other stakeholders, it will be difficult to judge whether LLM-based techniques provide any benefit to RE tasks compared with existing approaches. The F1 score is a useful measure of the performance of a model, but a poor indicator of the overall benefit in the application of the model. To date, much of the literature is based on measures of model performance rather than benefits from application. This is not unexpected given the relative infancy of LLMs applied to RE, but studies going forward should identify specific application use cases that enable comparisons between NLP and traditional methods on the basis of outcomes and relative benefits. In addition to the F1 score to assess the relative performance advantages of the model, the measures of performance for the RE tasks themselves should be a part of the assessment of the relative benefits provided by the application of NLP methods.

This article and its observations do not come without limitations: in contrast to Sonbol et al. (2022) and Zhao et al. (2021), we did not carry out a systematic review of LLMs applied to RE tasks, and as a result might have missed work that does not align with the challenges we identified in the cited research. However, given that the LLM applications to RE are still a small field, we believe it is unlikely that we missed any key publications not aligned with the observations made so far. The list of possible research directions to overcome the challenges is also likely to be

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

incomplete – and we look forward to seeing other directions mentioned in future research as well.

Acknowledgments

We would like to thank the editor and two anonymous reviewers for their feedback and insight which helped to improve this article.

Financial support

This work was financially supported by an industry collaboration with Airbus.

References

Ajagbe, M. & **Zhao, L.** 2022 Retraining a BERT model for transfer learning in requirements engineering: A preliminary study. In *2022 IEEE 30th International Requirements Engineering Conference (RE)*, pp. 309–315. IEEE.

Arora, C., **Grundy, J.** & **Abdelrazek, M.** 2023 Advancing requirements engineering through generative AI: Assessing the role of LLMs. Preprint, arXiv:2310.13976.

Aurum, A. & **Wohlin, C.** 2005 Requirements engineering: Setting the context. In: Aurum, A., Wohlin, C. (eds) *Engineering and Managing Software Requirements*, Berlin, Heidelberg: Springer. https://doi.org/10.1007/3-540-28244-0_1.

Austin, J., **Odena, A.**, **Nye, M.**, **Bosma, M.**, **Michalewski, H.**, **Dohan, D.**, **Jiang, E.**, **Cai, C.**, **Terry, M.** & **Le, Q.** 2021 Program synthesis with large language models. Preprint, arXiv:2108.07732.

Bajaj, D., **Goel, A.**, **Gupta, SC.** & **Batra, H.** 2022 MUCE: A multilingual use case model extractor using GPT-3. *International Journal of Information Technology* **14**(3), 1543–1554.

Berquand, A., **Darm, P.** & **Riccardi, A.** 2021 SpaceTransformers: Language modeling for space systems. *IEEE Access* **9**, 133111–133122.

Berry, D., **Gacitua, R.**, **Sawyer, P.** & **Tjong, S. F.** 2012 The case for dumb requirements engineering tools. In *Requirements Engineering: Foundation for Software Quality. REFSQ 2012*. Lecture Notes in Computer Science, Vol. **7195**, pp. 211–217.

Bruel, J.-M., **Ebersold, S.**, **Galinier, F.**, **Mazzara, M.**, **Naumchev, A.** & **Meyer, B.** 2021 The role of formalism in system requirements. *ACM Computing Surveys (CSUR)* **54**(5), 1–36.

Chami, M., **Zoghbi, C.** & **Bruel, J.-M.** 2019 A first step towards AI for MBSE: Generating a part of SysML models from text using AI. In INCOSE Artificial Intelligence for SystemsEngineering: 2019 Conference Proceedings, 1st Ed. INCOSE.

Chapman, W. W., **Nadkarni, P. M.**, **Hirschman, L.**, **D'avolio, L. W.**, **Savova, G. K.** & **Uzuner, O.** 2011 Overcoming barriers to NLP for clinical text: The role of shared tasks and the need for additional creative solutions. *Journal of the American Medical Informatics Association* **18**(5), 540–543.

Cleland-Huang, J., **Settimi, R.**, **Zou, X.** & **Solc, P.** 2007 Automated classification of non-functional requirements. *Requirements Engineering* **12**, 103–120.

Cosler, M., **Hahn, C.**, **Mendoza, D.**, **Schmitt, F.** & **Trippel, C.** 2023 Nl2spec: Interactively translating unstructured natural language to temporal logics with large language models. Preprint, arXiv:2303.04864.

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

Dalpiaz, F., **Dell'Anna, D.**, **Aydemir, F. B.** & **Çevikol, S.** 2019 Requirements classification with interpretable machine learning and dependency parsing. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pp. 142–152.

Dalpiaz, F., **Ferrari, A.**, **Franch, X.** & **Palomares, C.** 2018 Natural language processing for requirements engineering: The best is yet to come, *IEEE Software* **35**(5), 115–119.

Dawood, O. S. & **Sahraoui, A.-E.-K.** 2017 From requirements engineering to UML using natural language processing – Survey study. *European Journal of Industrial Engineering* **2**(1), 44–50.

de Araújo, A. F. & **Marcacini, R. M.** 2021 RE-BERT: Automatic extraction of software requirements from app reviews using BERT language model. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pp. 1321–1327.

Deeptimahanti, D. K. & **Sanyal, R.** 2011 Semi-automatic generation of UML models from natural language requirements. In *Proceedings of the 4th India Software Engineering Conference*, pp. 165–174.

Deshpande, G., **Sheikhi, B.**, **Chakka, S.**, **Zotegouon, D. L.**, **Masahati, M. N.** & **Ruhe, G.** 2021 Is BERT the new silver bullet? – An empirical investigation of requirements dependency classification. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pp. 136–145.

Devlin, J., **Chang, M.-W.**, **Lee, K.** & **Toutanova, K.** 2018 BERT: Pre-training of deep bidirectional transformers for language understanding. Preprint, arXiv:1810.04805.

Diamantopoulos, T., **Roth, M.**, **Symeonidis, A.** & **Klein, E.** 2017 Software requirements as an application domain for natural language processing. *Language Resources and Evaluation* **51**, 495–524.

Dick, J. & **Llorens, J.** 2012 Using statement-level templates to improve the quality of requirements. In *24th International Conference on Software & Systems Engineering and Their Applications* 1–11.

Draeger, L. 2023 Application of machine-learning-based natural language processing and transformer-based semantic similarity for requirements engineering automation. Master's thesis, Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Aachen, Germany.

Ferrari, A., **Spagnolo, G. O.** & **Gnesi, S.** 2017 PURE: A dataset of public requirements documents. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 502–505. IEEE.

Fischbach, J., **Frattini, J.** & **Vogelsang, A.** 2021. CIRA: A tool for the automatic detection of causal relationships in requirements artifacts. Preprint, arXiv:2103.06768.

Grishman, R. & **Sundheim, B. M.** 1996 Message Understanding Conference-6: A brief history. In *Proceedings of the 16th conference on computational linguistics* 466–471.

Gururangan, S., **Marasović, A.**, **Swayamdipta, S.**, **Lo, K.**, **Beltagy, I.**, **Downey, D.** & **Smith, N. A.** 2020 Don't stop pretraining: Adapt language models to domains and tasks. Preprint, arXiv:2004.10964.

Hahn, C., **Schmitt, F.**, **Tillman, J. J.**, **Metzger, N.**, **Siber, J.** & **Finkbeiner, B.** 2022 Formal specifications from natural language. Preprint, arXiv:2206.01962.

Hall, B., **Fiedor, J.** & **Jeppu, Y.** 2020 Model integrated decomposition and assisted specification (MIDAS). In *INCOSE International Symposium*, Vol. **30**, pp. 821–841. Wiley Online Library.

He, J., **Bartocci, E.**, **Ničković, D.**, **Isakovic, H.** & **Grosu, R.** 2022 DeepSTL: From english requirements to signal temporal logic. In *Proceedings of the 44th International Conference on Software Engineering*, pp. 610–622.

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

Hey, T., **Keim, J.**, **Koziolek, A.** & **Tichy, W. F.** 2020 NoRBERT: Transfer learning for requirements classification. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pp. 169–179. IEEE.

Hickey, A. M. & **Davis, A. M.** 2004 A unified model of requirements elicitation, *Journal of Management Information Systems* **20**(4), 65–84.

Hirshorn, S. R., **Voss, L. D.** & **Bromley, L. K.** 2017 NASA Systems Engineering Handbook. Technical report, NASA.

Howard, J. & **Ruder, S.** 2018 Universal language model fine-tuning for text classification. Preprint, arXiv:1801.06146.

Hull, E., **Jackson, K.** & **Dick, J.** 2005 *Requirements Engineering*. Springer.

INCOSE. 2015 *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. John Wiley & Sons.

Kici, D., **Malik, G.**, **Cevik, M.**, **Parikh, D.** & **Basar, A.** 2021 A BERT-based transfer learning approach to text classification on software requirements specifications. In *Canadian Conference on AI*, Vol. **1**, p. 042077.

Kof, L. 2005 Natural language processing: Mature enough for requirements documents analysis?. In *International Conference on Application of Natural Language to Information Systems*, pp. 91–102. Springer.

Kurtanović, Z. & **Maalej, W.** 2017 Automatically classifying functional and non-functional requirements using supervised machine learning. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 490–495. IEEE.

Laplante, P. A. & **Kassab, M.** 2022 *Requirements Engineering for Software and Systems*, Auerbach Publications.

Li, J., **Sun, A.**, **Han, J.** & **Li, C.** 2022 A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering* **34**(1), 50–70.

Lim, S. C. 2022 A case for pre-trained language models in systems engineering. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Lin, J., **Liu, Y.**, **Zeng, Q.**, **Jiang, M.** & **Cleland-Huang, J.** 2021 Traceability transformed: Generating more accurate links with pre-trained bert models. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pp. 324–335. IEEE.

Lucassen, G., **Dalpiaz, F.**, **van der Werf, J. M. E.** & **Brinkkemper, S.** 2016 Improving agile requirements: The Quality User Story framework and tool. *Requirements engineering* **21**, 383–403.

Luitel, D., **Hassani, S.** & **Sabetzadeh, M.** 2023 Using language models for enhancing the completeness of natural-language requirements. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pp. 87–104. Springer.

Malik, G., **Cevik, M.**, **Parikh, D.** & **Basar, A.** 2022 Identifying the requirement conflicts in SRS documents using transformer-based sentence embeddings. Preprint, arXiv: 2206.13690.

Manning, C. D. 2022 Human language understanding & reasoning, *Daedalus* **151**(2), 127–138.

Mavin, A., **Wilkinson, P.**, **Harwood, A.** & **Novak, M.** 2009 Easy approach to requirements syntax (EARS). In *2009 17th IEEE International Requirements Engineering Conference*, pp. 317–322.

Norheim, J., **Lim, S. C.**, **Kerbrat, A.** & **Rebentisch, A.** 2022 Methods for extracting structured data from engineering requirements using natural language processing. In *Proceedings of the 13th Complex Systems Design and Management Conference* 1–15.

Radford, A., **Narasimhan, K.**, **Salimans, T.** & **Sutskever, I.** 2018 Improving language understanding by generative pre-training.

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press

Rajan, A. & **Wahl, T.** 2013 *CESAR-Cost Efficient Methods and Processes for Safety-Relevant Embedded Systems*. Springer.

Rastogi, A., **Zang, X.**, **Sunkara, S.**, **Gupta, R.** & **Khaitan, P.** 2020 Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. **34**, pp. 8689–8696.

Reinhartz-Berger, I. & **Kemelman, M.** 2020 Extracting core requirements for software product lines. *Requirements Engineering* **25**, 47–65.

Rupp, C. 2014 Requirements templates – The blueprint of your requirement. https://www.sophist.de/publikationen/requirements-engineering-und-management/.

Ryan, K. 1993 The role of natural language in requirements engineering. In *[1993] Proceedings of the IEEE International Symposium on Requirements Engineering*, pp. 240–242. IEEE.

Sang, E. F. & **De Meulder, F.** 2003 Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. Preprint, arXiv:cs/0306050.

Soares, L. B., **FitzGerald, N.**, **Ling, J.** & **Kwiatkowski, T.** 2019. Matching the blanks: Distributional similarity for relation learning. Preprint, arXiv:1906.03158.

Sonbol, R., **Rebdawi, G.** & **Ghneim, N.** 2022 The use of NLP-based text representation techniques to support requirement engineering tasks: A systematic mapping review. *IEEE Access* **10**, 62811–62830.

Tikayat Ray, A., **Cole, B. F.**, **Pinon Fischer, O. J.**, **White, R. T.** & **Mavris, D. N.** 2023 aeroBERT-classifier: Classification of aerospace requirements using BERT. *Aerospace* **10**(3), 279.

van Lamsweerde, A. 2009 *Requirements Engineering: From System Goals to UML Models to Software Specifications*, John Wiley & Sons, Ltd.

Vaswani, A., **Shazeer, N.**, **Parmar, N.**, **Uszkoreit, J.**, **Jones, L.**, **Gomez, A. N.**, **Kaiser, L.** & **Polosukhin, I.** 2017. Attention is all you need. *Advances in Neural Information Processing Systems* **30**, 5998–6008.

Zhao, L., **Alhoshan, W.**, **Ferrari, A.**, **Letsholo, K. J.**, **Ajagbe, M. A.**, **Chioasca, E.-V.** & **Batista-Navarro, R. T.** 2021. Natural language processing for requirements engineering: A systematic mapping study. *ACM Computing Surveys* **54**(3), 1–41.

Zhu, Q. & **Luo, J.** 2022. Generative pre-trained transformer for design concept generation: An exploration. In '*Proceedings of the Design Society*, Vol. **2**, pp. 1825–1834. Cambridge University Press.

https://doi.org/10.1017/dsj.2024.8 Published online by Cambridge University Press