CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Natural Language Processing and Large Language Models for Automation of Compliance Tracing

Master's Thesis in Computer science and engineering

Maximilian Forsell
Eric Erlandsson Hollgren

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025

Master's Thesis 2025

Natural Language Processing and Large Language Models for Automation of Compliance Tracing

Maximilian Forsell
Eric Erlandsson Hollgren

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2025

Natural Language Processing and Large Language Models for Automation of Compliance Tracing
Maximilian Forsell
Eric Erlandsson Hollgren

Supervisor: Jan Bosch, Department of Computer Science and Engineering
Second supervisor: Helena Holmström Olsson, Department of Computer Science and Media Technology, Malmö University
Examiner: Christian Berger, Department of Computer Science and Engineering

Master's Thesis 2025
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2025

Maximilian Forsell
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Eric Erlandsson Hollgren
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Compliance is a costly and time-consuming task that most, if not all, firms must perform. As such, automating parts of the compliance process could be highly valuable. This thesis aims to investigate challenges faced by European software-intensive firms in their compliance processes, identify automation opportunities, and develop a Natural Language Processing- and Large Language Model-based software artifact to automate compliance tracing between company guidelines and normative requirements. The thesis followed the Design Science Research approach, and as such, the research was conducted in close collaboration with industry practitioners. The challenges and automation opportunities were identified together with seven interviewees from four different companies, and the final software artifact, dubbed *TraceAlign*, was developed and evaluated in focus groups with a total of twelve unique participants from two companies. The identified challenges ranged from organizational- and management-related to specifics inherent to the specifications of normative requirements. Automation opportunities related mainly to the management of requirements, company guidelines, and compliance evidence, of which this thesis focuses specifically on the task of compliance tracing of company guidelines to normative requirements. The final software artifact, *TraceAlign*, was considered to be time- and cost-saving by the focus group participants, but could perhaps be made more accurate. We conclude that there are many challenges with compliance that could potentially be automated using Natural Language Processing and Large Language Models.

Keywords: Compliance, Design Science Research, Large language model, Natural language processing, Artificial Intelligence, Knowledge graph, Requirements tracing

Acknowledgements

The authors would like to gratefully acknowledge the help of supervisors Jan Bosch and Helena Holmström Olsson. Without their help, valuable feedback and connections, the thesis would not have been possible. The helpful feedback from the thesis' examiner, Christian Berger, is moreover gratefully acknowledged. The authors would also like to thank Software Center for their contributions to the thesis through connections, workshops and presentations. Finally, the authors want to thank all interviewees and focus group participants for their time and valuable insights.

Maximilian Forsell & Eric Erlandsson Hollgren, Gothenburg, June 2025

Contents

List of Figures

List of Tables

1

Introduction

Compliance with regulations and standards is important for companies developing software-intensive systems. As the regulatory landscape evolves, ensuring compliance becomes increasingly labor-intensive and resource-demanding. This chapter introduces the context, motivation, and scope of the thesis, presenting the central problem of compliance tracing and the potential for automation using recent advances in Natural Language Processing (NLP) and Large Language Models (LLMs). It outlines the thesis's purpose, significance, contributions, and research questions.

1.1 Problem Statement

Compliance is *"the act of obeying a law or rule, especially one that controls a particular industry or type of work"* [2] or more specifically *"a firm's effort to ensure that it and its agents adhere to legal and regulatory requirements, industry practice, and the firm's internal policies and norms"* [3]. Compliance is a practice that every company either actively or passively works with, including companies that develop software-intensive systems. The research presented in this thesis was conducted in close collaboration with companies in Software Center. As such, the focus has been placed on such software-intensive companies. Further, software-intensive companies usually have to comply with several different regulations (e.g., ISO 26262 [4], IEC 62443-4-1 [5], ISO 21434 [6], Cyber Resilience Act [7], General Data Protection Regulation (GDPR) [8], etc.) and are therefore of particular importance to this study. Moreover, considering the authors' background in computer science, these companies were deemed the most suitable fit for this research.

Broadly, compliance can be divided into two categories: mandatory compliance, such as regulatory compliance, and voluntary compliance, such as compliance with a standard.

Regulatory compliance (or legal compliance) is a compulsory activity for all firms since non-compliance is, by definition, against the law [9]. Non-compliance with regulations can thus result in various legal actions being taken against the firm, including fines upwards of billions of euros [10]. New regulations are adopted over time, and old ones are amended, leading to a constantly changing and evolving field of regulations that firms must comply with. While some regulations are industry-

specific, many affect firms more broadly. For example, the Data Act [11] dictates rules for how data may be accessed and used within the European Union (EU), and the Artificial Intelligence (AI) Act [12] regulates the use of AI within the EU. Regulations such as these are of particular importance to software-intensive companies.

The main emphasis of this thesis is not on regulatory compliance but rather on standards compliance, a type of voluntary compliance. Complying with standards can be a source of competitive advantage as it demonstrates to both suppliers and buyers that an organization has high-quality products and processes [13]. Standards compliance can moreover be necessary for suppliers as many larger organizations require their suppliers to be certified according to certain standards [13]. Further, standards are occasionally created to comply with a regulation, and can thus help a firm stay compliant with this regulation [13]. One example of this is the standard ISO 21434, which enables firms to stay compliant with the UN Regulation No. 155 - Cybersecurity and cybersecurity management system (R155) [14], which will be elaborated more upon in Section 3.2.3. In general, standards are found in almost every industry, however, this thesis focuses specifically on standards affecting software-intensive companies in telecommunications, automotive, cybersecurity, and water solutions. In keeping with the terminology introduced by Castellanos Ardila et al. [15], the term "normative requirements" will hereby be used to refer to any document which an organization may need to comply with, e.g., standards, regulations, or policies.

Ensuring compliance is the firm's responsibility and is costly due to its labor-intensive nature [16]. Moreover, as more time and resources are spent on compliance, less time and resources are available for other customer value-creating activities. Thus, automation of compliance would be of great importance to many firms.

Complete automation of the problem is complex since any misinterpretations and non-compliance could result in large fines. While partial automation has been attempted in the past (see e.g., Castellanos Ardila et al. [15]), recent developments in NLP, namely LLMs, can potentially automate some of the more labor-intensive parts. LLMs could allow for near-human-level interpretation of normative requirements, which enables more fine-grained interpretations than before. Therefore, LLMs present new possibilities for partial automation of the compliance process.

Different firms work in various ways concerning compliance, which means that a naive one-size-fits-all approach is not feasible without first gathering and identifying common general challenges faced by companies. To the best of the authors' knowledge, few previous scholarly publications capture this information. Thus, there is a need to fill that gap and develop a software artifact that can partially automate the process for several firms.

Compliance processes in firms are multifaceted and involve various steps, roles, and methods in different stages. Completely automating the entire process and every different part and step is thus infeasible. Many standards specifically state that parts of the compliance process must be manually verified by a human, making it

de facto impossible to automate the entire process.

This thesis was carried out in close collaboration with industry partners. Building on a literature review of state-of-the-art NLP and LLM approaches to compliance automation, it specifically addresses the problem of *tracing* company guidelines to normative requirements. The solution presented is generally applicable, however, it is evaluated on three standards in the field of safety-critical systems, namely ISO 26262, ISO 21434, and IEC 62443-4-1.

Tracing (also referred to as tracking or mapping) is generally an essential part of staying compliant. It refers to the act of tracing specifically which normative requirement(s) a piece of compliance evidence is related to. For example, a sub-clause in a standard might state that there must exist explicit requirements specifying the type of encryption used in the end product. Then, when a manager or auditor wants to assess if the product is compliant with this sub-clause, they want to be able to trace that normative requirement all the way to a concrete requirement on the product. This tracing goes both ways and can, in a general scenario, be a many-to-many mapping. There are, in general, several middle steps between the actual compliance evidence and the normative requirements, for example, company policies, company guidelines, functional requirements, etc. Tracing should then be possible through all of these middle steps.

1.2 Purpose of the Study

The purpose of this thesis is to identify challenges with compliance for software-intensive companies and investigate an NLP- and LLM-driven approach to trace compliance of company guidelines toward normative requirements. The thesis produced both knowledge and software artifacts. The knowledge artifact consists of new insights into the challenges software-intensive companies face with compliance, as well as the effectiveness of using LLMs to automate compliance tracing. The software artifact is a tool for automating compliance tracing from normative requirements to company guidelines, which was evaluated on three safety-critical standards. Throughout this thesis, “compliance tracing” will refer to “compliance tracing from normative requirements to company guidelines”.

1.3 Significance of the Study

This thesis is particularly significant for the software-intensive systems domain, where companies operate under strict and complex regulatory environments, e.g., ISO 26262, IEC 62443-4-1, and ISO 21434. In these contexts, normative requirements are typically formulated in broad and general terms to accommodate a wide range of stakeholders and use cases [17]. As a result, the task of interpreting and implementing these requirements is left to individual companies. This opens the door to different interpretations, which can cause misalignment between development teams and regulatory bodies [17]. Such inconsistencies may lead to delays,

costly rework, or legal repercussions.

As is illustrated by the interviews conducted during this thesis, automating parts of the compliance process would be time- and cost-saving for many firms, particularly for firms developing software-intensive systems. There is a growing literature focusing on this subject, leveraging the latest developments in the field of NLP, in particular LLMs (see Section 4.2).

This study contributes both to the general literature on challenges within compliance for software-intensive firms and to the literature on leveraging NLP and LLMs for automating compliance. The former contribution consists of investigating the current state of compliance in European software-intensive firms and a potential solution for automating some part of it. The latter contribution consists of designing, implementing, and evaluating an NLP- and LLM-based approach to compliance tracing.

1.4 Research Questions

The thesis has three research questions (RQs):

RQ1: What are the challenges that software-intensive companies experience in complying with normative requirements?

This RQ aims to investigate the current landscape of compliance for software-intensive companies. The goal was to collect data on what challenges exist.

RQ2: What are the opportunities to address these challenges by automating parts of the compliance process using LLMs?

The subject of compliance is large and complex. Given the time and resources of a master's thesis, not all of the challenges regarding the automation of compliance can be solved. With this RQ the aim is to find specific parts of challenges that are feasible to automate.

RQ3: To what extent can an AI-system, relying on LLM-technology, sufficiently automate part of the compliance process in a firm?

Using the results of *RQ2*, the aim is to produce a software artifact that can automate some part of the compliance process at a firm. By "sufficiently automate" the authors refer to the fact that the software artifact should be value-creating for at least some firms, meaning that by implementing the software artifact, the firm can achieve cost-reductions, time-reductions, or accuracy-improvements compared to their current compliance process. The performance of the produced software artifact was evaluated together with domain experts at the partnering firms.

1.5 Contributions

This thesis makes three key contributions:

This thesis contributes to addressing the lack of research on the general compliance challenges faced by practitioners in European software-intensive firms. To bridge this gap, the study systematically gathered and analyzed interview data using thematic analysis. The thematic analysis resulted in five themes encompassing various challenges firms face in complying with normative requirements:

- "Challenges in the Organization" captures challenges related to coordination and communication between departments in small and large firms. It also captures the drawbacks of having a central compliance group, as experienced by some larger firms, and the drawbacks of lacking a central compliance group, as experienced by some smaller firms.

- "Challenges when Managing Global Markets" captures challenges that are present due to a firm's global reach. In particular, global firms must find which normative requirements are present on each market, determine which normative requirements are the most restrictive, and manage the trade-offs between the cost of complying with a normative requirement and the value of that market.

- "Challenges Inherent to the Specifications of the Normative Requirements" captures challenges that arise from how regulations and standards are written. Outdated, abstract, uncertain, and overlapping normative requirements require costly and time-consuming work from firms.

- "Challenges with Maturity of the Compliance Process" captures what challenges are faced by companies that are used to working with compliance (mature firms) versus what problems are faced by firms not used to compliance (less mature firms). Mature firms struggle with adapting to new and disruptive normative requirements, whereas less mature firms also struggle with having to create and integrate compliance processes, retrofitting processes and products, and the high resource cost of complying with a new normative requirement.

- "Challenges Related to Requirements" captures challenges related to tracing requirements, building and tracing guidelines for the requirements, and verifying that normative requirements have been correctly broken down.

Second, the thesis identifies several automation opportunities for compliance based on the interviews and a literature review. In total, interviewees identified eleven automation opportunities, most of which are related to tracing and verifying requirements as well as compliance evidence. The literature review revealed eight different fields of research that attempted to automate compliance using NLP and LLM methods. The thesis focused on automating three of the interviewee-identified opportunities based on several works from the literature review.

The third and final contribution is an automatic, NLP- and LLM-driven approach for automatic tracing in the compliance process, referred to as *TraceAlign*. The proposed software artifact works by using embeddings and Natural Language Inference (NLI) to automatically trace company guidelines to a set of normative requirements and then leverages LLMs to verify whether the trace is correct. *TraceAlign* builds upon the earlier works of Kruiper et al. [18] and Elluri et al. [19] by structuring the normative requirements in a knowledge graph and uses embeddings and NLI-models similarly to that of Hua et al. [20] to make traces. Moreover, *TraceAlign* leverages instruction-tuned Generative Pretrained Transformers (GPTs) to verify the correctness of traces, which appears to be a novel contribution to the field. Previous attempts leveraged fine-tuned Bi-directional Encoder Representations from Transformers (BERT)-models which required both training data and an exhaustive evaluation of all possible mappings [21], as opposed to *TraceAlign* which requires no training data and the number of possible mappings are reduced by the embedding and NLI-model.

The software artifact was created and validated in close collaboration with industry experts, focusing specifically on standards (ISO 26262, ISO 21434, and IEC 62443-4-1) compliance for safety-critical systems. Through focus groups, the authors demonstrate that the proposed software artifact can partially automate the tracing steps in compliance and thus make these steps faster and cheaper, albeit possibly less accurate.

1.6 Summary

This thesis explores how LLMs can be used to support the compliance process in software-intensive companies, focusing on the automation of compliance tracing between company guidelines and normative requirements. The study is motivated by the complexity and cost of manual compliance processes. It aims to identify common challenges across companies and develop a practical, partially automated solution, combining results from industry interviews with technical experimentation and evaluation.

The upcoming chapters are organized as follows:

- Chapter 2 introduces background on software-intensive systems and companies, compliance, NLP, LLMs, and knowledge graphs.
- Chapter 3 outlines the research methodology, including the Design Science Research (DSR) approach.
- Chapter 4 presents the answers to *RQ1* and *RQ2* based on interview answers and a literature review.
- Chapter 5 details the design and implementation of the proposed software artifact, *TraceAlign*.

- Chapter 6 presents the evaluation of *TraceAlign* and answers *RQ3*.
- Chapter 7 discusses implications, limitations, and future work.
- Chapter 8 concludes the thesis by summarizing key contributions.

2

Background

This section briefly introduces some of the background to the thesis, namely what a software-intensive company is, what compliance is, why it matters, and why companies might want to automate it. Some background on Machine Learning (ML) and NLP, as well as knowledge graphs, will also be presented.

2.1 Software-Intensive Systems and Companies

A software-intensive system is defined by van de Laar et. al [22] as *"systems with significant mechanical, optical, electrical, and other components, but where software accounts for a large part of the value and the development effort"*. Accordingly, this definition is used in the thesis. It encompasses both embedded software and software executed on machines outside of the core system, but provides services and value to the core system [22].

The authors define a software-intensive company as a company that develops and invests in systems and products that are software-intensive. Our definition is closely related to that of Findik and Tansel [23]. Findik and Tansel define a software-intensive firm as *"Firms that invest heavily in software products or services"*.

Companies working with embedded systems in telecom, automotive, defense, security, and manufacturing have started to complement their previous business models of producing and selling hardware products with software-driven services [24]. This shift reflects the growing strategic importance of software as an enabler of innovation and competitive advantage, even in previously hardware-focused sectors. Research shows that firms with a higher degree of software intensity tend to be more innovative and are more favorably perceived by the market, highlighting the trend of software becoming central to value creation in the manufacturing sector [25].

As companies working with embedded systems are becoming more software-intensive, it enables the continuous development of new features to their services and products [24]. An example from the automotive industry is over-the-air software updates, which can deliver updates to, for instance, the infotainment system and the driver assistance systems without the need to get the vehicle serviced. This shift also introduces a need for continuous compliance, as normative requirements must be

addressed dynamically throughout the product lifecycle rather than only at release.

Companies developing embedded systems face complexities due to the nature of the software-intensive systems. Embedded systems are often safety-critical, meaning that failures can have serious consequences. Therefore, every software update requires thorough verification and validation to ensure compliance with normative requirements, making frequent deployments non-trivial. Furthermore, embedded systems are often closely coupled with specialized hardware, which constrains what can be updated remotely and increases the risk of unintended side effects. Additional challenges include long product lifecycles and the need for real-time performance guarantees. These factors combine to make software-intensive companies interesting to study.

2.2 Compliance

This section provides insights into compliance with regulations and standards.

2.2.1 Regulatory Compliance

Regulations are necessary for the well-functioning of the economy by addressing various market failures and internalizing otherwise external social costs [26]. These regulations include, but are not limited to, EU regulations such as GDPR, the Data Act, and the Cyber Resilience Act. However, complying with these regulations, especially if they are poorly designed and implemented, can be a major cost for firms and can *"cause serious economic distortions that lower economic growth or GDP, damage investment and competitiveness and reduce entrepreneurship"* [27].

Regulatory compliance presents a high and increasing cost for many firms. As of 2023, reports indicate that one-third of firms in finance expect their compliance team to grow, and nearly three-quarters of firms expect to spend an increasing amount of time liaising and communicating with regulators and exchanges [28]. Similar observations have been made in other industries, i.e., that regulatory compliance accounts for a significant portion of firms' wage bills and is increasing over time [16]. Therefore, it stands to reason that the software industry is also being affected by this increasing trend, in particular in light of GDPR. 88% of global firms estimate that GDPR compliance costs alone reach more than $1 million annually, while 40% of them spend more than $10 million [29]. As of 2014, regulatory compliance costs accounted for 1.3%-3.3% of the total wage bill of the average U.S. firm, varying greatly with industry [16].

Some earlier attempts at automating regulatory compliance can be found. There exists software that streamlines and "automates" some parts of the regulatory compliance process, however, these frameworks generally focus on compiling data and monitoring processes. Integration of NLP tools such as LLMs is less common but can be found in academic literature (as reviewed in Section 4.2). Many of these solutions focus on extracting information from, e.g., data processing activities or

privacy policies to check for compliance against GDPR [30], [31].

2.2.2 Standards Compliance

Standards are voluntary, industry-developed guidelines, unlike mandatory government regulations. The adoption of standards ensures that companies within their respective field do not have to reinvent the wheel [32]. The purpose of standards is to ensure that products and services are safe, reliable, and of high quality, thereby increasing consumer confidence and contributing to a safer world [33], [32]. Firms choose to adopt and follow standards for many different reasons. One commonly cited (although disputed reason) is that firms comply with standards to in turn be compliant with regulatory requirements [13]. Other reasons include seeing it as an opportunity to gain a competitive advantage, gaining a first mover advantage, gaining entry into markets with strong competitors, to secure and reduce costs of international sales, to grow sales, and to improve the overall quality management of the firm [13].

Moreover, the adoption of standards allows companies to employ the most recent state-of-the-art solutions within their respective fields [32]. Long-term, this enables the company to build efficiency and reliability. However, short term, there is a big upfront cost in changing the current processes and products, as well as receiving the certification [13]. The benefits of following standards come from improving internal processes and building long-term competitive advantages [34]. Regulatory compliance, on the other hand, is primarily driven by the need to prevent external market failures and avoid penalties.

2.2.3 Product vs. Process Compliance

Both mandatory (regulatory) and voluntary (standards) compliance can be divided into “process compliance” and “product compliance”. These two types of compliance, although often interwoven, can and should be treated as different types of compliance. Product compliance focuses on whether or not a product and its properties adhere to certain rules, e.g., if a computer program has the correct level of encryption. Process compliance, on the other hand, focuses on the compliance of the processes during the development of the product, e.g., if the development was properly documented or sufficient safety analyses were conducted [35]. Figure 2.1 illustrates a typology of different normative requirements. The focus of this thesis is on process standards.

2.2.4 Safety-Critical Standards

Safety-critical systems are systems whose failure could result in significant harm to people, the environment, or property [36]. Safety-critical standards are here defined as a specialized subset of process standards that define structured approaches for developing safety-critical systems. The safety-critical standards included in this thesis are ISO 21434 [6], ISO 26262 [4], and IEC 62443-4-1 [5]. These standards are designed to ensure that rigorous and auditable processes are followed throughout

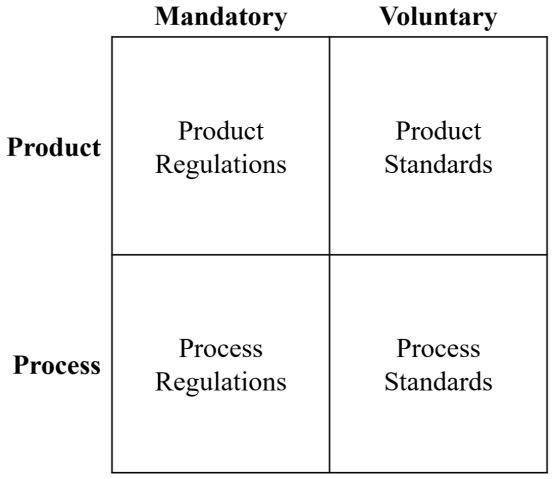

Figure 2.1: Typology of normative requirements.

the system lifecycle, from requirements engineering and design to implementation, verification, and maintenance. Their primary objective is to reduce the risk of malfunctioning behavior and to demonstrate that safety has been systematically addressed and validated. Safety-critical standards typically span both organizational capabilities and technical practices.

ISO 26262 [4] is a functional safety standard specifically tailored for electrical and electronic (E/E) systems in series production road vehicles. It provides a framework for identifying and mitigating hazards caused by malfunctioning behavior of safety-related E/E systems. The standard defines both technical and process requirements to guide the safe development, integration, and modification of such systems across their lifecycle. The standard aims to integrate functional safety into organization-specific development processes.

ISO 21434 [6] is a cybersecurity standard for E/E systems in series production road vehicles, covering all lifecycle stages. It defines a framework for managing cybersecurity risk, emphasizing both organizational processes and technical considerations. The standard establishes common terminology and structured processes for identifying, assessing, and mitigating cybersecurity threats throughout the development and integration of automotive systems and their components.

IEC 62443-4-1 [5] defines requirements for a secure product development lifecycle for industrial automation and control systems. It specifies process-oriented practices that product suppliers must follow to ensure that cybersecurity is systematically addressed throughout product development and maintenance. The goal of the standard is to embed cybersecurity into the development culture and ensure that products are built with security in mind from the outset. IEC 62443-4-1 is part of the broader IEC 62443 series, which provides a comprehensive framework for securing industrial systems across different roles and lifecycle phases.

2.2.5 Crosswalks

Crosswalks, as defined by Tupsamudre et al. [37], are mappings across several layers of compliance documents from regulations, harmonized controls (standards), all the way down to specific software implementations. Although not specifically stated in the original paper, company guidelines, policies, and processes that impose requirements on the final product could be included in the end-to-end mapping, as illustrated in Figure 2.2.

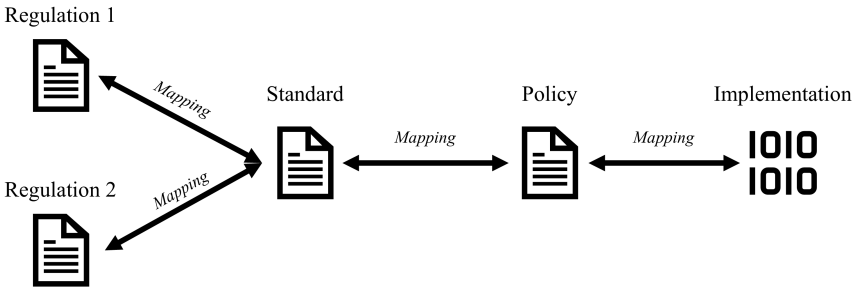

Figure 2.2: An example of a four-layer crosswalk from regulation to implementation.

Tupsamudre et al. [37] also present a formal definition. Let X be a compliance document where $X = \{x_1, x_2, ..., x_m\}$ and $x_i \; \forall \; i = 1, .., m$ are requirements. Then, let X_1 be the first-layer compliance document (the regulation in Figure 2.2) and X_n be the final-layer, i.e., code (the implementation in Figure 2.2). Then, an end-to-end crosswalk C exists if there is a mapping $X_1 \rightarrow X_2 \rightarrow ... \rightarrow X_n$. These mappings are important for ensuring that the implemented product is compliant with the highest-level normative requirements. Further, with the continuous development in software-intensive companies, there is a need for *"live crosswalks"* [37], i.e., crosswalks that are updated as the normative requirements and software implementations change.

2.3 Natural Language Processing

As stated, this article focuses on how NLP, specifically LLMs, can contribute to automating compliance. As such, a brief background on key concepts in NLP is warranted.

2.3.1 Word Embeddings

A precursor to understanding word embeddings is understanding tokenization. Tokenization is the act of turning documents, sentences, or words into tokens, which essentially means mapping the text into a pre-defined vocabulary of characters/-words [38]. E.g., words like "New York" may be mapped into a token "new york", "I'm" may be mapped into "i" and "am", and "documentation" may be mapped into "document-" and "-ation". In other words, the text is broken down into smaller pre-defined pieces. These tokens may be defined by hand in a rule-based manner, but more commonly, they are learned using an ML-based approach.

Word embeddings are then, in short, mappings from tokens into a vector space [38]. Generally, one distinguishes between sparse embeddings and dense embeddings.

2.3.1.1 Sparse Embeddings

Sparse word embeddings are, as the name implies, embeddings where most values for each vector are 0 [38]. Sparse vector embeddings can be created in several ways, where the general approach is to count words and how often they occur in the same context. For example, the simplest way of creating sparse word embeddings would be to count for each word how often it occurs next to other words in a corpus of text documents. Then, one constructs a so-called term-term co-occurrence matrix, where each word can be represented as a vector with the length of the total vocabulary with counts for each time they occur next to another word. Naturally this means that words that occur in similar contexts, i.e. have some semantic similarity, get similar vector representations. See Figure 2.3 for an example of a term-term co-occurrence matrix.

I love compliance. Compliance is fun. What is compliance?

⇓

	I	love	compliance	is	fun	what	.	?
I	0	1	0	0	0	0	0	0
love	1	0	1	0	0	0	0	0
compliance	0	1	0	1	0	0	1	1
is	0	0	1	0	1	1	0	0
fun	0	0	0	1	0	0	1	0
what	0	0	0	1	0	0	0	0
.	0	0	1	0	1	0	0	0
?	0	0	1	0	0	0	0	0

Figure 2.3: An example term-term co-occurrence matrix built from three sentences.

A more popular and sophisticated approach is term frequency-inverse document frequency (tf-idf) weightings [38]. Tf-idf weightings for a term are computed by first counting the frequency of the term t in a document d (term frequency) and then multiplying this by the inverse document frequency. The inverse document frequency is then defined as the total number of documents considered divided by the number of documents the term occurs. The intuition behind tf-idf is that words that occur in the same documents have something in common, however, some words occur in almost all documents (“and”, “the”, etc.), which is why the inverse document frequency is used to reduce the importance of these words.

Although sparse vector embeddings are by and large outdated, they remain a popular alternative in the literature. Tf-idf representations are particularly frequent in certain domains, see Section 4.2.

2.3.1.2 Dense Embeddings

Dense word embeddings do on the other hand not create sparse vectors, but rather smaller and denser vectors. The embeddings can generally be computed in two ways, statically (using e.g. word2vec) or dynamically (using e.g. an LLM) [38]. The key behind creating these embeddings is, however, that they are learned, not computed. In essence, dense word embeddings can be created by letting the weights of the first layer in an ML model trained on any type of word prediction task be the embeddings. Then, running text can be used in a self-supervised way to train this model, making the process of creating dense word embeddings relatively cheap with regards to data.

Generally speaking, dense word embeddings work better in every NLP task [38]. As such, the embeddings used in the software artifact produced in this thesis are always dense unless otherwise stated.

2.3.1.3 Semantic Similarity

Word embeddings are the standard way of representing words in NLP [38], and for good reason. Based on the way embeddings are computed, semantically similar words (i.e., words that occur in similar contexts) have vector representations that are similar to one another. Thus, if one has a database of documents that have been embedded, one can look up documents that are similar to one another, which is a key part of the retriever in Retrieval-Augmented Generation (RAG), see Section 2.3.3. First, a similarity measure (distance function) must be defined.

The standard definition of similarity between word embeddings is cosine similarity [38]. Cosine similarity is generally preferred as a distance measure in NLP settings since it does not favor long vectors over short vectors (unlike many other distance measures).

Cosine similarity between vectors $\mathbf{v}$ and $\mathbf{w}$ is defined as [38]:

$$\text{cosine}(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{v} \cdot \mathbf{w}}{|\mathbf{v}||\mathbf{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

Unless otherwise stated, cosine similarity is always used in this thesis to measure the semantic similarity between words and documents.

2.3.2 Large Language Models

LLMs is a collective term for any kind of language model that is large, i.e., has many parameters, as first introduced by Brown et al. [39]. Language models, in turn, are defined as any model that assigns a probability to a piece of unseen text based on its training data [40]. One key step in moving from simple language models to what would nowadays be referred to as LLMs is the introduction of the transformer architecture in 2017 by Google [41]. The transformer architecture, which will be dealt with in the proceeding section, is a precursor to the two most influential LLM-architectures to date, BERT [42] and GPTs [43].

2.3.2.1 Transformers

As mentioned, a language model is a model that assigns a probability to an unseen piece of text based on its training data. This means that any type of ML model can be a language model; however, most often, one thinks of language models as neural networks, as these are by far the most popular.

Neural networks are most easily introduced using mathematical notation. Let $\mathbf{x} = [x_1, ..., x_n]^T$ be an input vector and $\mathbf{y} = [y_1, ..., y_m]^T$ be an output vector. In the case of language models, both the input vector $\mathbf{x}$ and output vector $\mathbf{y}$ are generally vector representations of words (or more accurately, tokens). Then, a one-layer neural network is simply the following function [44]:

$$\mathbf{y} = h(\mathbf{W}\mathbf{x} + \mathbf{b})$$

where $h(\cdot)$ is a so-called activation function such as logistic ($h(z) = \frac{1}{1+e^{-z}}$) or ReLU ($h(z) = \max(0, z)$), $\mathbf{W} \in \mathbb{R}^{n \times m}$ is a weight matrix with learned weights and $b \in \mathbb{R}^n$ is the so-called bias, i.e. a vector of weights. A general neural network of depth L is then simply several of these functions stacked with varying dimensions and so-called hidden units (q) in between each layer [44]:

$$\begin{aligned}
\mathbf{q}^{(1)} &= h(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) \\
\mathbf{q}^{(2)} &= h(\mathbf{W}^{(2)}\mathbf{q}^{(\mathbf{1})} + \mathbf{b}^{(2)}) \\
\mathbf{q}^{(3)} &= h(\mathbf{W}^{(3)}\mathbf{q}^{(\mathbf{2})} + \mathbf{b}^{(3)}) \\
&\vdots \\
\mathbf{q}^{(L-1)} &= h(\mathbf{W}^{(L-1)}\mathbf{q}^{(\mathbf{L-2})} + \mathbf{b}^{(L-1)}) \\
\mathbf{y} &= h(\mathbf{W}^{(L)}\mathbf{q}^{(\mathbf{L-1})} + \mathbf{b}^{(L)})
\end{aligned}$$

Predicting only one output word based on one input word is however limiting. Generally, one wants to predict the next (or several next words) based on a sequence of input words. For these applications, where the data is sequential, recurrent neural networks are generally used. In short, these models can process a sequence of input vectors, $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(t)}$ of variable lengths [45]. The exact formulation will not be explained in detail here, however it is important to note two developments in recurrent neural networks, namely the Gated Recurrent Unit (GRU) [46] and Long-Short Term Memory (LSTM) [47]. In essence, both of these architectures aim at mitigating the vanishing gradient problem, i.e., that when sequences become too long, the gradients during training of the system become too small and the weights are updated poorly. Both architectures notably solve this by focusing only on certain parts of the sequences when updating.

The transformer architecture [41] develops this idea further by introducing what is referred to as attention functions or mechanisms. The original paper uses what is referred to as scaled dot product attention, which means that each transformer layer

learns three weight matrices: the query weights $\mathbf{W}^Q$, the key weights $\mathbf{W}^K$, and the value weights $\mathbf{W}^V$, which are used to compute query, key and value matrices:

$$\mathbf{Q} = \mathbf{X}_{query}\mathbf{W}^Q$$
$$\mathbf{K} = \mathbf{X}_{key}\mathbf{W}^K$$
$$\mathbf{V} = \mathbf{X}_{value}\mathbf{W}^V$$

where, in self-attention mechanisms, $X_{query} = X_{key} = X_{value}$. These matrices are simply the embedded input text, i.e. $X \in \mathbb{R}^{L \times d}$, where L is the length of the sequence and d is the number of dimensions in the embeddings. The attention function is then [41]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$$

where softmax is the activation function $\text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$, and d_k is the size of the key matrix. One can think of this function as allowing the neural network to learn which parts of the input text it should focus on in different settings, i.e., it learns how to weigh different tokens based on the context and their position. For brevity, some key concepts have been left out, namely masking and multihead attention. In short, it is common to mask the attention such that the network may only focus on the current token and all previous tokens (this is done in GPT models, but not in BERT models), which is referred to as causal masking. Multi-head attention simply means that each transformer model has several of these attention heads.

The transformer architecture, along with embeddings, lays the foundation for modern LLMs. As mentioned earlier, two types of LLMs are by far the most influential in the literature: BERT models and GPT models. The artifact produced in this thesis relies entirely on GPT models, however, BERT models are a popular, lightweight alternative, which is commonly used in the literature (see Section 4.2). As such, a brief review of both models is warranted.

2.3.2.2 Bi-directional Encoder Representations from Transformers

The original BERT model [42] was an encoder-only model trained on both the left and right context of each token (i.e., no causal masking). What this means in practice is that the BERT model only learns representations of words, and it is trained on predicting masked words based on context provided on either side as well as predicting whether two sentences are sequential. To use the model for any other task, such as question answering, language inference, etc., one must add a final layer fine-tuned on that task. Several works have also improved on the original model, all collectively referred to as BERT models. Important examples include RoBERTa [48] (a larger version) and DistilBERT [49] (a smaller version).

The architectural structure of BERT models has important ramifications for their performance on certain tasks. First of all, since the model learns words based on both left and right context (i.e., based on both preceding and proceeding words), they are generally not used for generating long sequences of text, since this task only

gives the model the left context (i.e., the preceding words). Moreover, since it only learns representations of words, a non-finetuned BERT model will be effectively useless for any general task (unless that task is predicting masked words or next sentences). In other words, to use a BERT model, it is required to have at least some small sample data or a model that is already fine-tuned for that task. Both of these aspects are in stark contrast to the GPT models.

2.3.2.3 Generative Pre-trained Transformers

The GPT models feature a different architecture and were trained differently. The original GPT-model [43] used a masked Transformer-decoder architecture trained on a very large corpus of unlabeled text. Practically, this just means that the model is trained at predicting the next word in a sentence, given only the left context (all preceding words). Similarly to the BERT models, the original paper [43] suggested that the model could then be finetuned on specific tasks.

However, a few years later an important advancement was made, colloquially known as "few-shot learning" [39]. Essentially, it was discovered that when the model becomes large enough, it is enough to supply a few examples of what the model is supposed to do in a prompt, and it will be able to continue this pattern. In other words, these larger GPTs could solve general problems without fine-tuning. Another important improvement came in 2022, by Wei et al. [50], when the authors demonstrated the possibility to fine-tune a GPT specifically on a large labeled dataset of instruction-following tasks, referred to as "instruction tuning". These models could then solve general tasks without any examples whatsoever, besides a prompt, i.e., they are "zero-shot learners".

This thesis relies entirely on instruction-tuned GPTs for all AI-based tasks. Few-shot prompting techniques may still be utilized, however, since this can improve performance on certain tasks [51].

2.3.3 Retrieval-Augmented Generation

Traditionally, LLMs follow instructions based on a simple prompt provided by the user. The LLM then uses its *parametric knowledge* to follow these instructions, i.e., it follows the instruction based on previous training examples it was exposed to during instruction-tuning. For certain tasks requiring specialist knowledge, parametric knowledge might be insufficient since the LLM has not been exposed to these tasks during training. Then, one can introduce *non-parametric knowledge* in the prompt, a process often referred to as RAG [52]. A traditional RAG pipeline works by first embedding chunks of documents containing knowledge in a vector space using word embeddings. Then, when a user inputs a query, semantically close documents are retrieved from the vector space and inputted as context for the LLM in answering the question. Figure 2.4 illustrates a simple RAG pipeline.

To improve the accuracy of the retrieved results, several methods can be employed. For example, the embeddings of the query may be modified in several ways. A simple

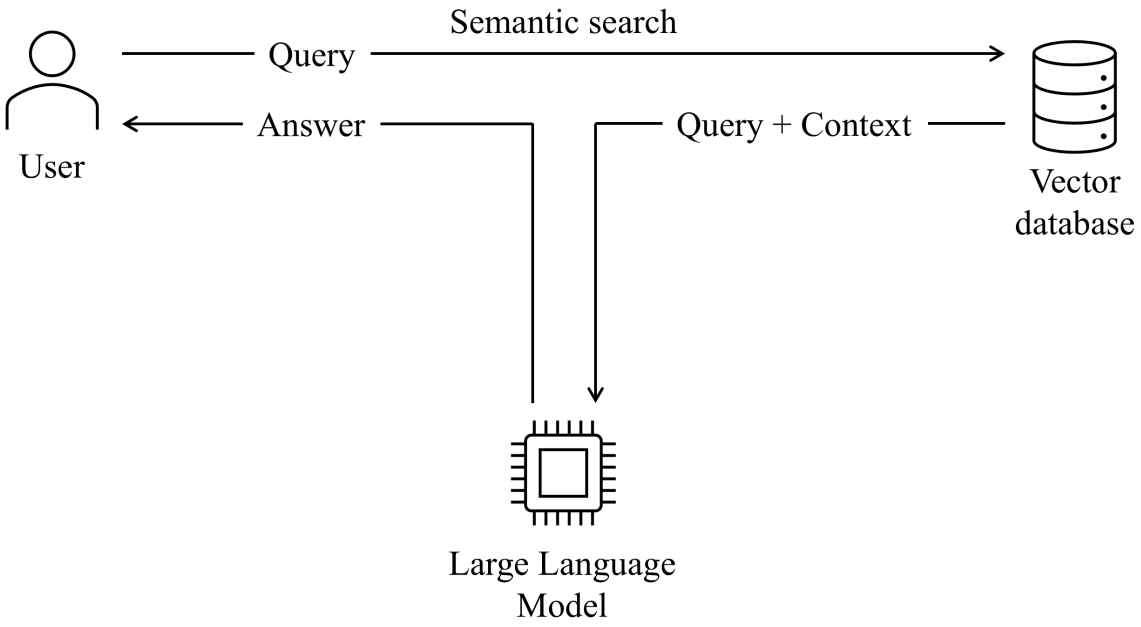

Figure 2.4: A simplified Retrieval-Augmented Generation pipeline.

approach would be to remove stop words before embedding or add definitions to rare or situation-specific terms to ensure that the embeddings of the query-text is closer to the embeddings of the text containing the correct knowledge. Recent approaches instead prompt an LLM with the query, and ask it to rewrite it in such a way that it is likely to have better embeddings [53]. One specific such technique is referred to as Hypothetical Document Embeddings (HyDE) [54], which entails prompting the LLM to rewrite the query as a hypothetical document which answers the question the query is looking for, which should then intuitively be worded (and embedded) similarly to any real answer to the question in the knowledge base. Importantly, the hypothetical document may include factual inaccuracies and false information, but this is not problematic since it is only used to retrieve the actual answer.

Besides modifying the embeddings of the query, it is also possible to evaluate the retrieved chunks of text against the query to determine which ones are best, a process often referred to as *reranking* [55]. Often, reranking is done by using a cross-encoder[1] model which has been trained on labeled pairs of query and answer passages such that it has learned which passages are good answers to queries and which are not, as done by Santhanam et al. [57] with the ColBERTv2 model.

Traditionally, reranking is done using models that have been trained on pairs of questions and answers. However, any model that takes two passages as input and outputs a score of some kind can be used as a reranker. One example would be to use a model trained for textual entailment, often referred to as NLI, i.e., a model that has been trained on pairs of sentences to determine if they are "entailed", "neutral"

[1]In short, cross-encoder models are similar to the BERT-models described above, however they can take as input two sentences/passages and output a prediction, e.g. if the right sentence contains the answer to the left sentence [56].

or “contradictory” [42].

2.4 Knowledge Graphs

A knowledge graph is a way to bridge the gap between real-world concepts and enterprise data management. Using a knowledge graph, users can bring relationships and real-world concepts to the forefront instead of relying on database schemas [58]. The knowledge graph consists of nodes and edges, which respectively represent real-world concepts and relationships that together make up the graph. This representation allows the graph to combine and integrate data that originate from several different sources [58]. Knowledge refers to concepts and the relationships between them being treated as core elements, capturing how domain users perceive and understand the world. A graph is a data structure based on nodes and edges that allows for the integration of data from diverse sources, ranging from unstructured to structured formats.

Figure 2.5 shows an example of a knowledge graph that illustrates some basic relationships between flora and fauna.

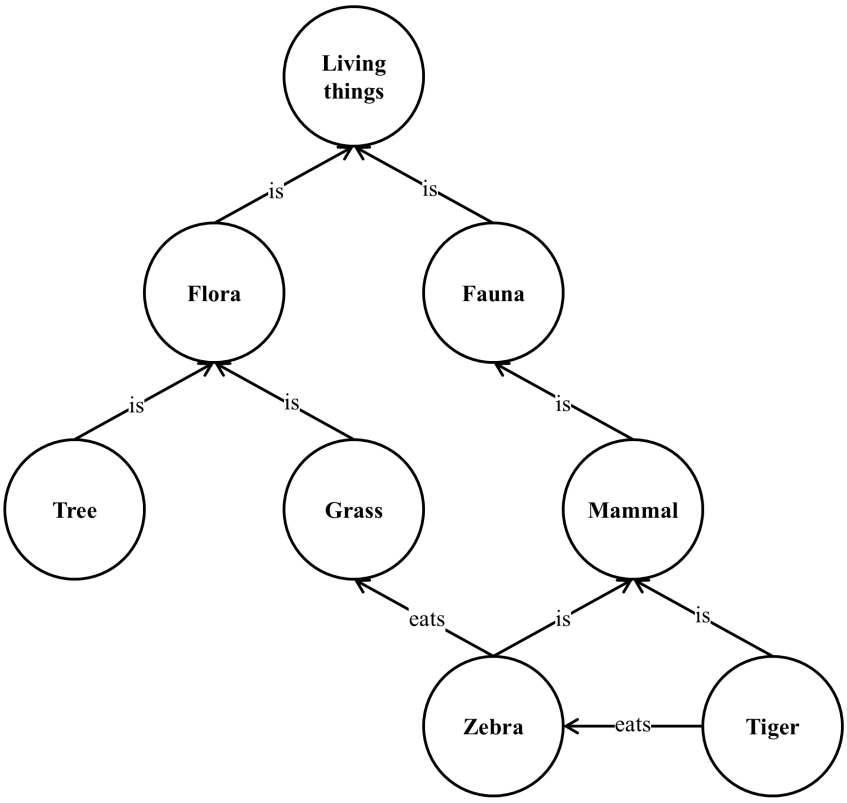

Figure 2.5: An example of a knowledge graph.

2.4.1 Property Graphs

A property graph can be used to construct a knowledge graph. The property graph model provides greater flexibility compared to directed edge-labeled graphs [59].

The reason for the increase in flexibility is the introduction of property-value pairs and labels to both nodes and edges within the property graph. Nodes and edges in a property graph are treated as structured objects [59]. One of the databases that implements property graphs is Neo4j, which was the graph database of choice for this thesis.

Figure 2.6 illustrates an example of a property graph with some basic properties.

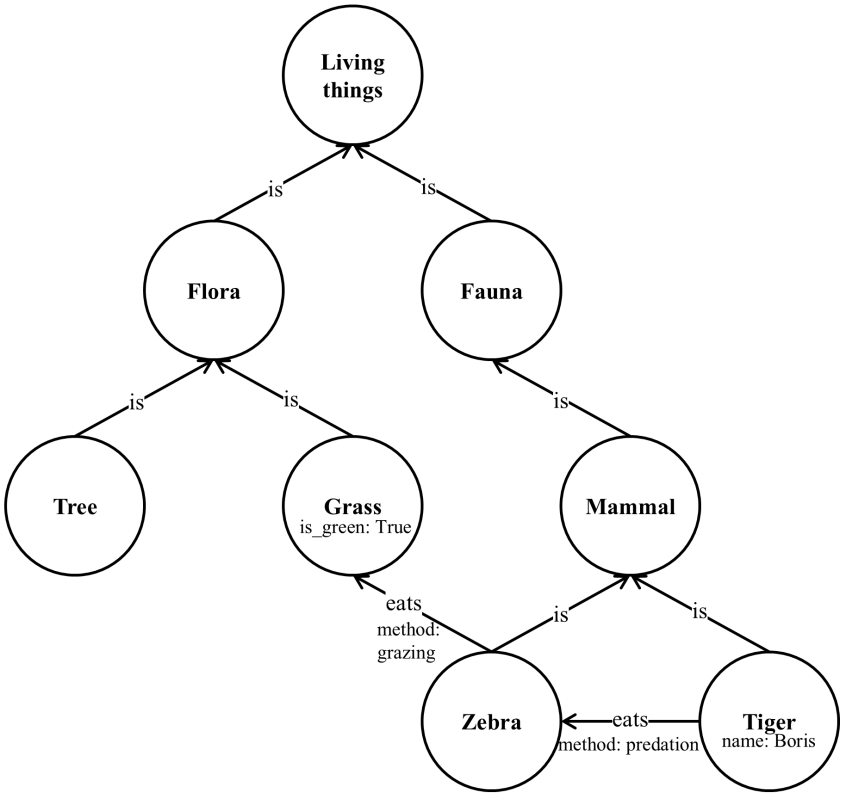

Figure 2.6: An example of a knowledge graph implemented as a property graph.

2.4.2 Ontologies

A knowledge graph may also incorporate an ontology as a way to organize and formally represent the knowledge graph data [58]. Generally, in the context of knowledge graphs, these ontologies or schemas consist of three main components [58]:

- *Classes*: an abstraction for creating collections of objects. E.g., the class "Tiger" may contain two instances, "tiger-1" and "tiger-2". Both classes and instances are then nodes in the graph.

- *Datatype properties*: relations between instances of classes and literals. E.g., "tiger-1" might have a "name", "Boris". Then, "tiger-1" is an instance, "name" is the datatype property and "Boris" is the literal. The datatype property is then typically an edge in the knowledge graph.

- *Object properties*: relations between instances of two classes. For example, the two tigers may be friends, represented by a "friend" object property, i.e. an edge between the two instances.

The ontology representation of a knowledge graph was not used in this thesis. However, it is a popular alternative in the literature (see Section 4.2), which warranted this brief review.

2.5 Summary

This chapter has briefly summarized some important concepts related to software-intensive companies, compliance, NLP, LLMs, and knowledge graphs. Importantly, it has described the characteristics of software-intensive companies and highlighted the importance of compliance for such firms. This thesis builds on the crosswalking concept by tracing between normative requirements and company guidelines, which is evaluated on three safety-critical process standards: ISO 21434, ISO 26262, and IEC 62443-4-1. The crosswalking is enabled by embeddings, NLI-rerankers, LLMs, and is structured as a property graph.

3

Research Methods

The thesis is an exploratory field study using the DSR framework. This section provides a detailed description of DSR and explains how the framework has been applied in the study. It also outlines and justifies the overall methodology for data collection and analysis.

3.1 Design Science Research

DSR is a problem-solving approach that seeks to enhance human knowledge by generating innovative artifacts as solutions to real-world problems [1]. Often, DSR is used for creating and evaluating IT artifacts intended to solve identified organizational problems [60]. It was used in this thesis to create an NLP and LLM-based software artifact to solve an identified compliance problem, namely, compliance tracing. The DSR approach has in recent years generated a surge of interest due to its potential to contribute to fostering the innovative capabilities of organizations [1], and has moreover been shown to create artifacts with significant economic and societal impact, particularly in information systems research [61], [62]. DSR is central in many different fields such as engineering, architecture, business, economics, and information technology, where it is used for creating innovative solutions to design problems [1].

The DSR approach was particularly appropriate for this thesis due to its emphasis on building and evaluating artifacts to address practical problems. While collaboration with industry practitioners is also common in other research approaches, such as Action Research, DSR is distinct in that the creation of an artifact is central to both the research process and its outcomes. Given that neither researcher had prior experience in the compliance field, close collaboration with practitioners was essential not only to identify relevant challenges but also to ensure the artifact addressed them effectively. The researchers collaborated with four software-intensive companies to identify key challenges (see Table 3.1), and subsequently developed and evaluated the artifact in close collaboration with two of these companies (B and D). This collaboration included multiple on-site visits and continuous communication throughout the development process.

Since the research was conducted in a specific, real-world setting and the purpose was

Company	Industry
A	Mobile telecommunications
B	Automotive
C	Cybersecurity
D	Water solutions

Table 3.1: Companies collaborated with throughout the thesis

to study a specific software engineering phenomenon (requirements and compliance tracing), this thesis was a field study [63]. According to Stol and Fitzgerald [63], a field study enables the authors to gain deep knowledge into a specific problem in a real-world scenario. However, this comes at the cost of low precision of measurement and low generalizability. Given the context and nature of this study, we determined that a field study leveraging the DSR framework represents the most appropriate approach. To investigate the RQs within this field study, the authors employed a combination of qualitative and quantitative methods: semi-structured interviews (see Section 3.2.1), literature review (see Section 3.2.2), factorial experiments (see Section 3.2.3), focus groups (see Section 3.2.4), thematic analysis (see Section 3.3.1), and logical argument (see Section 3.3.2).

DSR is generally done in cycles, each consisting of activities [1]. The research starts with activity 1, which is not a part of any cycle. This activity defines the problem and demonstrates its importance [1]. After the first activity is completed, the cycles begin. Each cycle contains four activities: First, the researchers define the objectives of the cycle to improve the artifact. Second, the researchers design and implement the new artifact. Third, the artifact is tested for solving the problem defined in activity 1. Fourth, an evaluation of the new artifact is done. Most cycles end at step 4 and begin a new cycle afterwards, apart from the last cycle, where a fifth activity is to communicate the findings. The different activities, workflow, and methods used during the cycles of this thesis are illustrated in Figure 3.1. The thesis adhered to the guidelines for conducting a design science master thesis by Knauss [64].

The proposed way of conducting DSR is that its done iteratively [64], [1]. A master thesis usually fits 3 cycles, where each respectively lasts about one month [64] and consequently, this thesis consisted of three cycles. In each of the cycles, it is expected that the artifact should improve. As such, each cycle must gather data for all the proposed RQs (*RQ1-3*) [64], [1]. However, the focus for each cycle shifts between the RQs.

The software artifact produced must thus be evaluated in one way or another in each cycle. It should be noted that there are generally trade-offs when evaluating an artifact, namely trade-offs between rigor, efficiency, and ethics [65]. In relation to this thesis, the main trade-off in evaluation was between rigor and efficiency. A rigorous evaluation of the produced artifact would require extensive testing and evaluation by experts in compliance and industry practitioners, however, this would also be very time- and cost-intensive. The thesis approached this trade-off by having

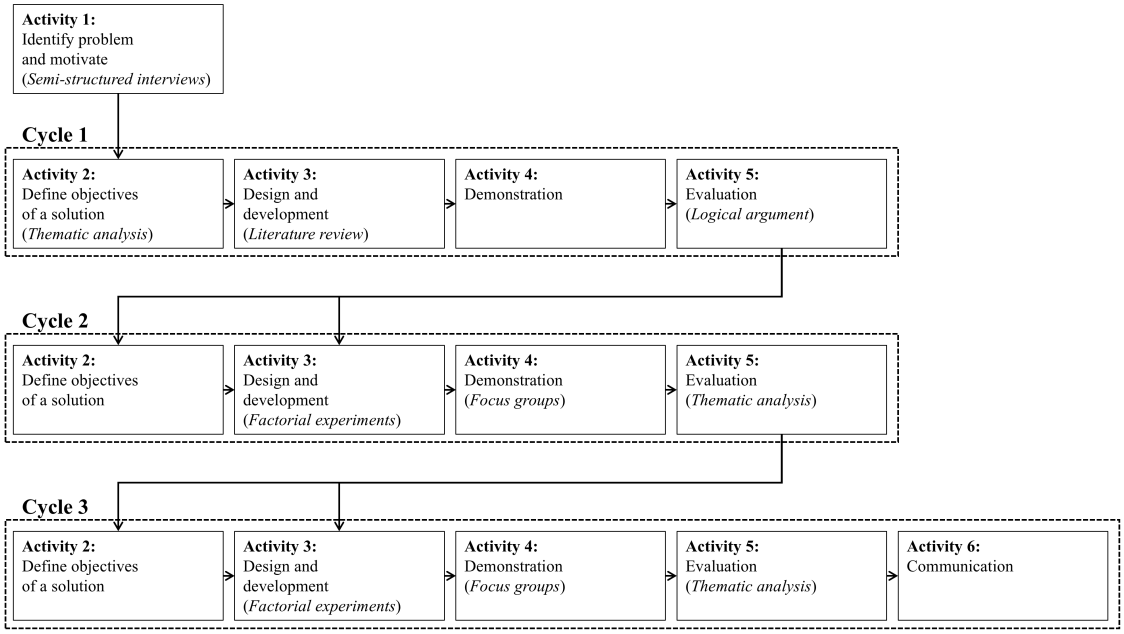

Figure 3.1: Activities and workflow of the thesis [1]. The italic text is the method used for that specific activity.

a less rigorous evaluation in the first cycle, to instead have a more rigorous evaluation in the second cycle, and in particular, the third and last cycle.

For each of the cycles shown in Figure 3.1 *Activity 3* illustrates the continuous development of the software artifact (*Tracealign)* as well as what inspired the design. As can be seen in Cycle 1, the literature review was the main source of design decisions when producing the first flowchart of the software artifact. During Cycle 2 and Cycle 3, the factorial experiments guided the design of the software artifact. The combination of the previous cycle's evaluation *(Activity 5)* together with continuous feedback from practitioners and the intermediary evaluation in each cycle's *Activity 3*, enabled the authors to design and develop the software artifact.

The following sections will describe the methods used in each of the cycles, divided into data collection and data analysis. Prior to any of the cycles, semi-structured interviews were conducted with industry practitioners working with compliance. The first cycle then defined the objectives of a solution based on the results from the thematic analysis of the interviews. Based on this analysis, a flowchart of a suggested solution was developed by the authors and validated internally using a logical argument. The second and third cycles collected data through focus groups and factorial experiments, which were analyzed and used for evaluation of the software artifact.

3.2 Data Collection

Data collection consisted of four different methods: semi-structured interviews, a literature review, factorial experiments, and focus groups.

3.2.1 Semi-Structured Interviews

This section presents *Activity 1* conducted before any cycles shown in Figure 3.1.

To answer *RQ1* and *RQ2*, interviews were conducted with compliance practitioners in software-intensive companies. The interviews were structured as expert interviews [66], a form of semi-structured interviews, which is the dominant form for qualitative interviews [67]. Semi-structured interviews (also referred to as semistandardized [68]) entail that the researchers create a formal list of open-ended questions, to be asked in a consistent and systematic order, but the interviewer is allowed (and in fact, should) digress from the questions and ask follow-up or unplanned questions [67], [68]. Expert interviews were used since the interviewees are of less interest than the expert knowledge they possess [66]. Thus, the interviewees were seen as representatives of a group [66]. The purpose of using expert interviews for this thesis was to explore and orient the field of compliance, especially the challenges with it and the potentials for automation.

The list of interview questions, i.e., the interview guide, used for the expert interviews can be found in Appendix A.3, along with the consent form and participant information sheet. These were prepared in accordance with Clarke and Braun [67]. An interview guide is particularly important when conducting expert interviews [66].

First, the main body of questions, meant to answer *RQ1* and *RQ2*, were brainstormed by the authors, independently of one another. Second, beginning and ending questions were created. Third, the main body of questions was reorganized to flow logically and grouped into main topics: “background information”, “challenges with the process”, “challenges with interpretation”, and “data and tools”. Finally, once a draft of the interview guide was done, each question was polished to ensure that they were correctly phrased, following Clarke and Braun [67], and potential follow-up questions were prepared, listed as sub-items in the guide.

The semi-structured expert interviews were conducted over a four week period and included seven experts working in four different companies. For a complete list of the anonymous interviewees and their corresponding companies, see Table 3.2. When possible, interviews were conducted face-to-face, as this is generally advantageous [67]. This was unfortunately only possible for Company B, and the other interviews were thus conducted in virtual meetings. Interviews were scheduled to take around one hour. All interviews were recorded and transcribed automatically using Microsoft Office.

To mitigate bias and capture different perspectives on the compliance process and its challenges, participants had different roles within the organizations. The process of choosing interviewees usually consisted of a contact person at an organization to whom the authors described the study and its purpose. The contact person then redirected the authors to professionals at the organization who could participate in the semi-structured interviews.

As seen in Table 3.2, the interviewees possess a diverse set of roles within the different

organizations. Many of the roles, such as: "R&D Manager", "Security architect", "Compliance manager", "Platform architect", and "Functional safety manager" are senior members of the organization, which was desirable as compliance requires high-level decisions. Moreover, the role "Functional safety team member" is also desirable since they work with the direct effects of the high-level compliance decisions. To capture even more fine-grained effects of compliance, it would have been desirable to interview engineers as well, but unfortunately, that was not possible.

Interviewee	**Role**	**Company**	**Date**	**Length of Interview**
Interviewee A1	Domain expert	Company A	21-01-2025	~30 min[1)]
Interviewee B1	Functional safety team member	Company B	22-01-2025	~1 hour
Interviewee B2	Functional safety manager	Company B	22-01-2025	~1 hour
Interviewee C1	R&D Manager	Company C	22-01-2025	~1 hour
Interviewee A1	Domain expert	Company A	31-01-2025	~30 min[1)]
Interviewee D1	Security architect	Company D	02-02-2025	~1 hour[2)]
Interviewee D2	Compliance manager	Company D	02-02-2025	~1 hour[2)]
Interviewee D3	Platform architect	Company D	02-02-2025	~1 hour[2)]

[1)] This interview was conducted in two parts.
[2)] The interview with Interviewee D1, D2 and D3 was conducted as one group interview. This was partly a necessity out of time considerations but also due to the interviewees possessing knowledge about different parts of the compliance process in the firm.

Table 3.2: Interviews conducted in Cycle 1

3.2.2 Literature Review

This section presents *Activity 3* conducted in Cycle 1, shown in Figure 3.1.

In accordance with Jesson [69], a traditional literature review was undertaken for the thesis to explain the current state-of-the-art in automating compliance using NLP while also establishing a research gap and answering *RQ2*. The literature review specifically takes the form of a state-of-the-art review [69], since it brings the reader up to date on the most recent related works. It moreover informed the design and development of the software artifact by guiding the authors in the field of compliance automation. The identified papers also illustrated which of the interviewee-identified automation opportunities were already covered in the literature.

The literature review was conducted to explore the current state-of-the-art in automating compliance. Of particular interest to this thesis is the prior work related to using NLP and LLMs for automating compliance. To find such work, the following search phrase was used:

(LLM* or "large language model*" or NLP or "natural language processing") and (complianc*)

The search phrase was input into Web of Science and ACM Digital Library, restricted to works published after 2019, i.e., in the range 2020-2025. The search matched based on title, abstract, and keywords.

The literature was analyzed to find a research gap and inspirations for how the software artifact could be designed (*RQ2*), implemented and evaluated. The important parts of each paper, i.e., the domain it covers, what problem it is trying to solve, the NLP/LLM methods used, and possibly how well the artifact performed, were briefly summarized. This is standard procedure in a traditional review [69]. Once each paper had been summarized, it was classified based on what problem it was trying to solve, and if necessary, papers were grouped if they attempted to solve the same problem, in a similar domain, using the same methodology.

3.2.3 Factorial Experiments

This section presents *Activity 3* conducted in Cycle 2 and Cycle 3, shown in Figure 3.1.

To supplement the qualitative results, experiments were run on a "labeled" dataset provided by one of the partnering companies (Company B).

Assessing the accuracy of the software artifact's tracing function was difficult, as this would have required a dataset with mappings from one document to another. An assessor would not generally produce such a dataset when assessing the compliance of a document or a company as a whole. However, Company B was able to provide a mapping from a regulation (R155) to a standard (ISO 21434) that had been meticulously constructed over 6 months by several experts and PhD students. While this dataset was not a perfect representation of the task the software artifact would typically handle, it was conceptually similar and served as a proxy for determining the accuracy of the artifact's tracing ability. Moreover, it enabled the authors to tune the software artifact and measure how well the artifact captured human intuition.

The purpose of supplementing the qualitative results with factorial experiments was a combination of establishing a baseline performance and identifying areas for improvement of the software artifact. These quantitative experiments were also used for validating the functionality of the software artifact and for closing the gap between the qualitative data and quantitative measurement

Data and possibilities for evaluating the software artifact were not abundant. As such, the quantitative experiments were designed to extract as much data as possible from the datasets.

Throughout various stages of development, the software artifact had different parameters and factors that affected performance and accuracy. In order to tune these parameters and determine the effect of varying the factors, factorial experiments were used [70].[1] When experiments included factors that were more qualitative

[1] In the context of tuning hyperparameters in ML, the terminology "hyperparameter grid search"

than quantitative (e.g., the type of embedding model used, or the type of LLM used), factorials with mixed levels were employed [70].

The outcomes of the experiments were evaluated using several metrics based on the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) predicted by the model. A TP refers to a trace that should have existed and was correctly created by the model. A TN refers to a trace that should not have existed and was correctly not created. An FP refers to a trace that should not have existed but was incorrectly created, while an FN refers to a trace that should have existed but was not created. These outcomes are used in the following metrics, which were used to quantitatively assess the performance of the software artifact:

$$\begin{aligned}
\text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN}, \\
\text{Recall} &= \frac{TP}{TP + FN}, \\
\text{Specificity} &= \frac{TN}{TN + FP}, \\
\text{Precision} &= \frac{TP}{TP + FP}, \\
F_\beta\text{-score} &= \frac{(1 + \beta^2) \cdot TP}{(1 + \beta^2) \cdot TP + \beta^2 \cdot FN + FP}
\end{aligned}$$

3.2.4 Focus Groups

The demonstrations (*Activity 4* in Figure 3.1) in Cycle 2 and 3 were done using focus groups, although for slightly different purposes and thus using slightly different approaches. In these cycles, data were collected to evaluate the current software artifact prototype.

In Cycle 2, the purpose of evaluation was equivalent to the fifth purpose listed by Venable et al. [65]: *"evaluate a designed artifact formatively to identify weaknesses and areas of improvement for an artifact under development"*. In other words, the purpose of the evaluation was to evaluate the current software artifact prototype to identify areas of improvement and refinement. In Cycle 3, the final cycle, the purpose of evaluation was the second one listed by Venable et al. [65]: *"evaluate the formalized knowledge about a designed artifact's utility for achieving its purpose"*.

In order to demonstrate the software artifact developed in each cycle, focus groups of people who work with compliance were gathered from the partnering companies. Focus groups are essentially any group discussion where the researchers are actively encouraging of and attentive to the group interaction [72]. Originally, focus groups stem from media and marketing research, where the focus has been explicitly on using group interactions that would have been less accessible in a non-group setting.

is more commonly used [71].

The focus groups aimed to get feedback on the artifact produced from previous cycles [72].

Focus groups in this study consisted of people with varying backgrounds and roles who, in one way or another, worked with compliance against normative requirements. Groups were homogeneous with respect to which company they worked for, but heterogeneous with respect to their roles and experiences with compliance. When conducting focus groups, it is generally important that the participants in each group share at least one important characteristic [73], which in this case was the company and normative requirements they worked with. All focus groups taken together were, however, heterogeneous with respect to the companies studied. Thus, each singular focus group allowed the researchers to study and analyze the perspectives and interactions of people with different roles and perspectives on compliance, whereas all focus groups as a whole allowed the researchers to contrast the needs and wants of different companies. Being able to analyze both within and between groups is important and standard when conducting focus groups [72]. Moreover, the focus groups always included new participants not present in any previous cycle to mitigate bias.

The full protocol used for demonstrating and evaluating the prototype in Cycle 2 can be found in Appendix B. In short, the authors briefly introduced how the software artifact worked, its intended use cases, and demonstrated some of its features. Questions to the audience were asked throughout the demonstration and also after, in a semi-structured way. Table 3.3 summarizes the six anonymous participants from the focus group conducted at Company B on 12-03-2025.

Participant	**Role**
Participant B1	Functional safety team member
Participant B2	Functional safety manager
Participant B3	Functional safety team member
Participant B4	Safety engineer
Participant B5	Safety engineer
Participant B6	Functional safety team member

Table 3.3: Focus group participants in Cycle 2

The full protocol used for demonstrating and evaluating the prototype in Cycle 3 can be found in Appendix C. Two focus groups were conducted, one in person at Company B on 2025-05-12 with six participants and one virtually at Company D on 2025-05-13 with four participants. Both focus groups took approximately one hour each. Table 3.4 summarizes the anonymous focus group participants.

Participant	Role
Participant B1	Functional safety team member
Participant B2	Functional safety manager
Participant B3	Functional safety team member
Participant B4	Safety engineer
Participant B7	Consultant
Participant B8	Consultant
Participant D1	Security architect
Participant D2	Compliance manager
Participant D5	Head of legal
Participant D6	Data scientist

Table 3.4: Focus group participants in Cycle 3

All focus groups took approximately one hour. The audio from all demonstration sessions was recorded and transcribed automatically using Microsoft Office.

3.3 Data Analysis

Two methods for analysis were employed in this study. First, thematic analysis was employed to analyze the results of the interviews and focus groups, albeit with slightly different approaches for each type of data. Second, a logical argument was carried out in the first cycle to validate the flowchart that was produced.

3.3.1 Thematic Analysis

This section presents *Activity 2* conducted in Cycle 1, *Activity 5* in Cycle 2, and *Activity 5* in Cycle 3, shown in Figure 3.1.

The interviews and focus groups were analyzed using inductive thematic analysis [74]. All interviews and focus groups conducted were transcribed and manually combed through by the authors to extract codes by analyzing answers at the sentence level. Codes were generally quotes or paraphrases from the interviewees that were found meaningful with regards to the RQs being investigated [74].

The extracted codes were then grouped into categories. Each category described the underlying meaning of the extracted codes. The methodology for the process is described by Alvinius et. al [74]. Finally, after categorizing the codes, the results were mapped onto themes as described in Alvinius et al. [74]. Each theme captured one or several categories and encoded the more general concepts. The extracted themes were reviewed one last time, and those without bearing to the RQs were not included in Chapter 4 and Chapter 6.

Table 3.5 shows some examples of quotes from interviewees, and how they were turned into codes, categories, and themes. Quotes in Swedish have been manually translated by the authors.

3.3.2 Logical Argument

This section presents *Activity 5* conducted in Cycle 1 as shown in Figure 3.1.

As discussed by Venable et al. [65], there are, in general five different purposes for evaluating a design theory, artifact, or hypothesis in DSR. In Cycle 1, the purpose of evaluation corresponds to the first purpose mentioned, i.e., to *"evaluate an instantiation of a designed artifact to establish its utility and efficacy (or lack thereof) for achieving its stated purpose"*. The type of evaluation used in Cycle 1 most closely resembles the "logical argument" presented in Peffers et al. [75].

Based on the literature review and interview analysis, the authors gained an understanding of the challenges firms currently face in their compliance processes, as well as the issues that have been addressed in existing research and the corresponding solutions. Using these insights, the authors brainstormed multiple potential solution candidates. The final selection was made based on the previously gathered data, ensuring that the chosen solution was both feasible within the given time frame and novel in relation to the identified literature.

In the first cycle, no working artifact was produced. Instead, the authors produced a conceptual idea that was represented by a flowchart. It was produced in an iterative fashion by first coming up with a solution, then evaluating it against the problems identified and methods used in the literature. Once a final flowchart had been created, the authors evaluated it internally by ensuring that it was both novel and able to solve one of the challenges found. Since no working artifact was developed, this was a type of *ex ante* evaluation [76].

3.4 Data Triangulation

The thesis collected data from various sources: qualitative data from semi-structured interviews and focus groups with industry practitioners; qualitative data from the literature review; and quantitative data from the factorial experiments. Using different data sources to validate findings is often referred to as data triangulation [77].

Data triangulation was used in this thesis to validate the correctness of the findings using various sources. For example, the automation opportunities identified by interviewees were compared with the general literature to determine which new automation opportunities had been found and which had already been automated. The focus group feedback was complemented by quantitative findings from the experiments, which gave different types of input for the researchers. The focus group feedback gave more general insight into what features might be missing and what traces the artifact incorrectly created, whereas quantitative data gave more precise insights into how the different components affected the results. Overall, the use of different data sources allowed the researchers to draw conclusions and validate the findings with more confidence than would have been possible using only one data source.

The qualitative data gave rich descriptions of the real-world scenarios in which the software artifact was situated. This data was generally not captured by the quantitative data. The quantitative data, however, captured more comprehensively the overall performance and accuracy of the software artifact in a way that could not be captured using qualitative data. The combination of these methods allowed the researchers to investigate when the findings from the data types aligned or differed.

3.5 Summary

This thesis used the DSR methodology structured into three cycles:

Before Cycle 1, semi-structured expert interviews with compliance practitioners were conducted, which were analyzed using thematic analysis to answer *RQ1* and *RQ2*. Cycle 1 then included a traditional state-of-the-art literature review to complement the answer to *RQ2*. This cycle produced a flowchart that was evaluated using a logical argument.

Cycle 2 implemented the flowchart from Cycle 1 into a functional software artifact, which was designed and developed with the help of factorial experiments and demonstrated in a focus group. The results from the focus group were analyzed using thematic analysis to guide the development of the final software artifact.

Cycle 3 improved upon the software artifact from Cycle 2 based on the results from the thematic analysis. The improved and final software artifact, *TraceAlign*, was designed and developed using factorial experiments. It was then demonstrated in focus groups, which were evaluated using thematic analysis to answer *RQ3*.

Quote	Code	Category	Theme
"So I think from a compliance point of view, the typical activities would be to help settles the scope of compliance of compliance or countries applications"	A typical activity would be to settle the cope of compliance or countries applications	Compliance affects the decision-making of firms	Parts of the Compliance Process
"All developers do not need or shouldn't be experts in all of these compliance issues"	Developers should not need to be experts in compliance issues	The compliance process requires support processes	Parts of the Compliance Process
"This standard [ISO 26262] is very fuzzy since it is meant to be followed by automotive in the entire world. (...) We need to do something called integration for example, but what that is it doesn't say exactly"	The standard is fuzzily written	Problems with interpreting normative requirements	Challenges Inherent to the Specifications of the Normative Requirements
"[When asked about how hard it is to interpret normative requirements] I would like to say that it is pretty hard."	It is pretty hard to interpret normative requirements	Problems with interpreting normative requirements	Challenges Inherent to the Specifications of the Normative Requirements
"It [ISO 26262] is hard to map directly into the organization"	Hard to map normative requirements into the organization	Problems with mapping the normative requirements into the organization	Challenges Inherent to the Specifications of the Normative Requirements

Table 3.5: Examples of quotes and their respective codes, categories, and themes

4

Compliance Challenges and Automation Opportunities

This chapter presents the answers to *RQ1* and *RQ2* based on the semi-structured interviews and the literature review. The answers are presented in sections based on themes identified by the authors during the thematic analysis performed in Cycle 1. The themes were derived based on groupings of quotes and paraphrases from industry practitioners.

4.1 Compliance Challenges

This section answers *RQ1*: *What are the challenges that software-intensive companies experience in complying with normative requirements?* The answer for *RQ1* is based on the results from the thematic analysis performed in Cycle 1.

4.1.1 Challenges in the Organization

Several of the identified challenges can be linked to the organizational structure of the firm trying to stay compliant. There are difficulties related to both larger and smaller firms,[1] albeit slightly different ones.

Larger firms are prone to challenges with coordination, hierarchies, and slow bureaucracies. There are often several different, semi-isolated departments working on different parts of the product. However, while the departments are semi-isolated, the product itself is not. Requirements on the product may come from several departments and thus require them to coordinate with one another to produce a safe, compliant product. This leads to departments having to proactively guess what requirements may come from other departments or design their requirements in such a way that they can accommodate many other solutions. This was emphasized by one interviewee: *"A hundred different groups are coming to place demands on us. No one has done it yet, so we have to guess what requirements they'll come with, and try to prepare a bit, so we have something that's probably roughly correct"*. This results in extra work for the employees.

[1] Small firms are here defined as having less than 500 employees.

Some companies have created a central group that aims to break down the normative requirements into more fine-grained requirements that can be used downstream in the organization. However, this group generally worked too slowly for the individual departments. The interviewees also expressed that the "fine-grained" requirements are generally not fine-grained enough for the engineers to use, and the department heads still need to manually interpret them, leading to a hierarchy of requirements being broken down.

Smaller firms generally face different challenges, mostly related to a lack of a compliance department and a lack of automation tools. As mentioned during the interviews by a manager in a relatively small firm: *"a compliance manager is not the first person you hire"*. As such, when smaller firms have trouble interpreting a standard, they must generally turn to costly consultants.

In both large and small firms, there are communication challenges that cause problems in the compliance process. Interviewees at larger firms expressed that while there is a central group for interpreting standards, it is not easy to communicate with them. The smaller firms instead answered that it was important that engineers invent the compliance process and managers provide the tools to support the said process. This was identified as a difficult coordination challenge.

4.1.2 Challenges when Managing Global Markets

This theme captures the challenges of companies that operate in global markets (and different vertical markets). When a company is to launch a product on several global markets, there exist different normative requirements that exert constraints on the product.

One challenge this imposes on the company is what normative requirements to comply with for each of the different global markets. As noted by one interviewee in the context of developing a new product: *"What are the standards we need to be compliant with?"*

Another challenge is to find which of the normative requirements are the most restrictive on the processes and demands of the product. Several interviewees highlight this challenge, for example: *"You aim for the limit values that are the strictest, so to speak"*, and: *"compliance requirements, if there are many, they are diverse and but they're also kind of like the lowest common denominator"* when discussing normative requirements on a global market.

For each product sold on several markets, the company has to weigh the cost of choosing the most restrictive normative requirements against the value brought by that market. This challenge was also brought to attention by an interviewee when discussing the restrictiveness of normative requirements on different markets: *"It can also drive a lot of cost. You might need certain materials or special solutions to meet the requirement, and it might not be economically sustainable to sell a product with that level of performance when it doesn't actually need to meet that standard."*

4.1.3 Challenges Inherent to the Specifications of the Normative Requirements

Companies face challenges aligning current software development practices with existing normative requirements. While these requirements have been updated over time, they often lag behind the pace of industry evolution. One interviewee highlighted that some normative requirements still fail to reflect how software is developed today: *"They [ISO] didn't really consider that the standard's foundation itself is around 15 years old, and technology has moved on. The way they saw systems back then doesn't fully apply anymore."*

Another challenge is the alignment between several normative requirements when applied to a product or process. As brought to attention by one interviewee: *"They [normative requirements] serve the same purpose but it is not always the case that they want to do things in the same way"*. This highlights that normative requirements often expect the same output but differ in the methods for producing that output, which imposes a challenge for the firms when deciding which method to use.

A last challenge is how to manage abstract and uncertain normative requirements. Several interviewees mention that there is a need for a compliance advisory. One interviewee expressed: *"Request for compliance advisory in the future because it can be very difficult to really nail it in terms of did we really satisfy this requirement or how did we interpret this"*, emphasizing that interpretations of the normative requirement can be challenging.

4.1.4 Challenges with Maturity of the Compliance Process

This theme encompasses challenges concerning compliance maturity for the participating companies. During the interviews, a distinction emerged between mature and less mature firms. Maturity was understood as the presence of an established compliance process, as opposed to firms still in the process of developing one.

Mature firms are generally less prone to compliance challenges than less mature firms. However, the adaptation of new normative requirements into current compliance processes poses a challenge, albeit a smaller challenge than for less mature firms. As an interviewee at a mature firm noted: *"That scenario [of having to comply with a new regulation] would be much worse if having a company where you don't have a practice for working (...) or at least have a relatively mature process framework"*.

Less mature firms face several challenges with the compliance process. The first challenge is that the firms have to create and integrate a complete compliance process into their current operations. In relation to how one firm worked with compliance, an interviewee mentioned that *"(...) If you start trying to sort something like this out, it makes a whole bunch of other things that may have been swept under the rug for many years painfully obvious"*, highlighting the challenge of integrating compliance into the current operations.

The second challenge for less mature firms encompasses retrofitting previous work to also be compliant with the adopted normative requirement. In many cases, development had already begun before the normative requirements were in place. As one participant expressed: *"(...) people have started without having anything in place. So now I guess you have to patch and redo some parts (...)"*. This highlights the challenge of aligning legacy operations and products with new normative requirements.

The last challenge for less mature firms is the high resource demand of adopting a new normative requirement, due to the lack of existing compliance processes. One interviewee at a less mature firm emphasized the inefficiencies with the lack of compliance processes as: *"(...) Unreasonable with the lead times and unpredictability"*. There is also a need for training and education as noted by an interviewee at another company: *"It's a matter of providing guidance, training and at the end of the day, a lot of education. But that is required"*.

4.1.5 Challenges Related to Requirements

This theme describes the challenges with requirement management noted in the interviews.

One challenge mentioned was the difficulty of tracing incomplete or loosely defined requirements. As one interviewee explained when talking about what current tools lack: *"Yeah, it's about keeping things together, like the traceability between requirements and all that, because that's a big part of it too. (...) As a person, even if the link isn't perfect, you can still understand that, yeah, it connects to that."*, emphasizing that current tools cannot trace incomplete requirements, which poses a challenge.

Ideally, developers should build great products and not be experts on compliance issues. This imposes the challenge of building and tracing comprehensive guidelines based on the normative requirements. As mentioned in an interview: *"The requirements from the standard (...) can be very complex to understand and then translate into guidelines (...) Developers do not need or shouldn't be experts in all of these compliance issues"*.

The last identified challenge was the breakdown of requirements. Specifically, retaining all the information at a higher abstraction level when breaking it down into lower-level requirements. One interviewee mentioned that: *"For the lower level requirements, when you break them down, you really need to cover the higher level requirement. It is easy to lose coverage"*, which highlights this challenge.

4.1.6 Summary

Companies face significant organizational and process-related challenges in achieving compliance. In large companies, compliance is difficult because departments work separately, central compliance teams are slow, and communication between

groups is often poor. Engineers often need to work proactively without waiting for centralized interpretation of normative requirements, leading to inefficiencies and inconsistent understanding of requirements. Smaller firms struggle due to the absence of dedicated compliance roles and automation, often needing to rely on external consultants. Both types of firms highlight the importance of compliance maturity. Those with prior experience integrate compliance more seamlessly, while less mature firms must build compliance processes from scratch, which is resource-intensive.

Process-wise, challenges include the manual and iterative nature of verification, the complexity of translating abstract normative requirements into concrete requirements and guidelines, and the difficulty of tracing these normative requirements throughout the system. Additionally, outdated or overlapping normative requirements create interpretation problems, especially when firms operate in multiple global markets with differing normative requirements.

Table 4.1 summarizes the challenges identified during the interviews. The challenge "Building and tracing comprehensive guidelines" (bolded in the table) is the central focus of this thesis.

Themes	Challenges
Challenges in the Organization	Coordination between departments working on the same product (*Large firms)*
	Inefficient central compliance group (*Large firms)*
	Lack of compliance department (*Small firms*)
	Communication challenges between departments
Challenges when Managing Global Markets	Finding the correct normative requirement for each market
	Finding the most restrictive normative requirement
	Managing the trade-off between cost of normative requirement and value of market
Challenges Inherent to the Specifications of the Normative Requirements	Normative requirements do not reflect the current software development landscape
	Alignment between normative requirements that output the same artifact but use different methods
	Managing abstract and uncertain normative requirements
Challenges with Maturity of the Compliance Process	Adaptation of new normative requirements (*Mature firms)*
	Creation of the compliance process and integration of it into the current operations (*Less mature firms)*
	Retrofitting processes and products that were created before the adoption of a normative requirement (*Less mature firms*)
	High resource demand when adopting a new normative requirement (*Less mature firms*)
Challenges Related to Requirements	Tracing incomplete or loosely defined requirements
	Building and tracing comprehensive guidelines
	Verifying information retention in breakdown of requirements

Table 4.1: Compliance challenges for software-intensive companies

4.2 Automation Opportunities

This section answers *RQ2*: *What are the opportunities to address these challenges by automating parts of the compliance process using LLMs?* The answer for *RQ2* is based on the answers to questions 2 and 3 in the interview guide (see Appendix A.3 in the section *Data and tools*) and is complemented by the literature review conducted in Cycle 1. The bolded opportunities for automation are the main focus for the software artifact in the thesis. Works from the literature review not related to any of the interviewee-identified automation opportunities have been omitted.[2]

4.2.1 Opportunities for Automatic Compliance Tracing

The first automation opportunity emerging from the interviews relates to the need for improved traceability and mapping between normative requirements, internal

[2]See Appendix D for the full literature review.

guidelines, and work products. Participants described the difficulty of tracing how specific normative requirements are addressed in internal processes, architectural models, and code. This challenge becomes even more complex when dealing with overlapping standards, where a single work product might contribute to compliance with several requirements. Interviewees proposed automation ideas such as:

1. Map written code to architecture models to show how technical components align with compliance objectives.
2. **Support querying across an integrated information model to trace compliance coverage.**
3. **Establish and maintain trace links between work products, internal guidelines, and external regulations.**
4. Generate justifications or evidence for how and where specific requirements have been satisfied.
5. Help manage reference structures that connect multiple regulations.

The literature on automatic compliance tracing is quite scarce, but there exists some prior work in these automation opportunities. The first, second, and third opportunities are related to the automatic generation of crosswalks in Agarwal et al. [21], the mapping of Common Weakness Enumerations to Common Vulnerabilities and Exposures in Turtianien and Costin [78], and the mappings from cybersecurity requirements to vendor-supplied features in Ameri et al. [79]. However, none of these works focus on including company guidelines in their methods, and all rely on fine-tuned BERT models, which require training data.

As for the fourth opportunity, generating compliance justification and evidence, there are several solutions. For example, some works propose using LLMs and few-shot learning to generate data processing activities and privacy practices to comply with data protection and privacy regulations [80] [81]. Similarly, Khakzad Shahandashti et al. [82] focus on using LLMs for automatically generating defeaters for assurance cases, i.e., structured arguments that allow one to determine if a system's non-functional requirements have been correctly implemented.

The fifth and last automation opportunity, managing reference structures, has not received much attention in the scholarly literature. However, Rahmani et al. [83] propose a conceptually-related system for gathering and summarizing references across project contracts, which could in practice be extended to regulations and standards in the general case.

4.2.2 Opportunities for Automatic Breakdown and Verification of Requirements

Another opportunity identified through the interviews relates to the breakdown and verification of requirements, especially when managing inputs from multiple

regulatory sources. Participants expressed the need for generic tools capable of handling requirements derived from various normative requirements, and highlighted the following opportunities for automation:

1. Translating high-level regulatory language into actionable requirements.
2. Transferring requirements across organizational, legislative, and process layers.
3. Automated breakdown of high-level regulations into actionable internal requirements
4. Assisted verification by interpreting requirement language and linking it to relevant evidence.
5. Real-time compliance feedback during development, rather than just end-of-cycle reporting.

Verification and breakdown of requirements, and compliance checking in general, have received quite a lot of attention in the scholarly literature, in particular in the construction and data privacy domains. Generally, methods in the construction domain leverage traditional NLP-tools to convert Building Information Modeling (BIM) models into annotated rules [84], [85], [86] or ontologies and knowledge graphs as a middle step between parsing the document and converting it to logical rules which can be used to check compliance [87], [88], [89], [90], [91], [92]. In the data privacy domain, a number of works focus on the automated completeness checking of data processing agreements, privacy policies and app permissions [31], [93], [94], [95], [96], [97], [98], [99], [100], [101], [102].

As for interpreting requirement language, there are a number of works that have leveraged LLMs for question-answering and paragraph location, see for example Xu et al. [103], Abualhaija et al. [104], [105], Deldari et al. [106], Sofat and Sodhi [107], and Seong et al. [108].

4.2.3 Opportunities for Automatic Compliance Structuring

The final opportunity identified in the interviews relates to the structuring of the data from the normative requirements. Participants emphasized the importance of codifying standards into structured formats to support scalable and efficient compliance tracing. They highlighted the opportunity of:

1. **Automatically codifying and structuring an unstructured standard (e.g, Office documents) into an ontology trace.**

A number of works in the literature review structured the compliance process as an ontology or a knowledge graph, which is also confirmed by Kotal et al. [109]. A knowledge graph-based approach applied to BIM is presented in Kruiper et al. [18], and a similar approach for GDPR compliance is used in Elluri et al. [19]. An ontology-based solution is presented in Javed et al. [110] for space software engi-

neering process compliance. Other simpler approaches to compliance structuring are presented in Zhou et al. [111] and Xue and Zhang [112], which use traditional NLP methods to convert natural language normative requirements into clearly annotated rules.

4.2.4 Summary

The interviews revealed several opportunities for addressing compliance challenges through automation using LLMs. Participants saw potential in tools that could automatically codify and structure unstructured standards into ontologies. They also highlighted the value of establishing and maintaining trace links between work products, internal guidelines, and external regulations. Furthermore, interviewees highlighted the opportunity to enable querying across an integrated information model, allowing stakeholders to trace compliance coverage more effectively.

Very few of the prior works in NLP/LLM-based automation of compliance have focused on developing a general method and framework that works for normative requirements in any domain. Furthermore, many of the prior works are based on training an ML system on a manually curated dataset in a specific field, and as such, these solutions might be less likely to generalize across fields.

With regards to compliance tracing, there is a lack of few-shot, zero-shot, and unsupervised approaches to automating this task, in particular creating crosswalks using GPTs. Current research in this field relies on inefficient pairwise comparisons to make traces, which would not work with LLMs and larger data sets. Although LLMs have been used in a number of works, these have been focusing on BERT-based models, with GPT-based models remaining relatively uncommon. GPTs have generally only been used in the field of question-answering and evidence generation, whereas the other challenges have been solved using traditional ML models trained on labeled datasets. Furthermore, LLMs could support more flexible output generation.

With the bolded opportunities in mind, the authors designed and developed a software artifact for NLP- and LLM-based compliance tracing automation, *TraceAlign*.

5

TraceAlign

This chapter presents the software artifact developed in this thesis: *TraceAlign*. The design of the artifact is directly informed by the underlined challenges and opportunities presented in Sections 4.1 and 4.2. The version presented in this chapter reflects the final iteration, which was evaluated in Cycle 3 to assess its utility, usability, and fit within real-world compliance processes. The final design of the software artifact was guided by insights gained from practitioner focus groups conducted in Cycle 2.

5.1 Problem Identification and Motivation

Tracing and maintaining well-written and comprehensive guidelines are at the source of *TraceAlign*. The challenge was brought to attention many times during interviews and focus groups as being a central, yet time-consuming and demanding task to do manually. All of the participating companies in this thesis build guidelines for their engineers to use in product development, and therefore the guidelines must be comprehensive and traceable to their respective normative requirements. These recurring observations made the challenge particularly relevant and impactful to address. Further, engineers produce evidence from guidelines, thus, the software artifact can easily be extended to trace evidence through the guidelines to normative requirements if the traces between guidelines and normative requirements exist and are correct.

Other challenges were also identified during the semi-structured interviews and are presented in Section 4.1. However, most of them are situated at an organizational level and are not readily addressable through the design of an IT artifact, making the selected challenge more suitable for technological intervention. Moreover, many of the other identified challenges were only present in some of the interviewed companies, and as such, a solution to them would not be value-creating for the entire group.

The most closely related research found was by Agarwal et al. [21], who developed a system for performing crosswalks between normative requirements and company policies. Their approach is based on an exhaustive search using a BERT model with pairwise comparisons between regulatory requirements and a standard. This prior work supports the feasibility of addressing similar challenges using NLP and

LLM-based techniques, validating the design of *TraceAlign*.

This research gap, together with industry practitioners expressing that the challenge is central for achieving complete compliance, motivates the need for a software artifact like *TraceAlign*.

5.2 Purpose and Use Cases of TraceAlign

The purpose of *TraceAlign* is to enable faster tracing from normative requirements to internal guidelines and persist the results in a knowledge graph. This tracing enables compliance by establishing machine-readable links between external normative requirements and an organization's internal controls or procedures. By automating and structuring traceability, *TraceAlign* aims to reduce manual effort and support continuous compliance monitoring. However, it does not attempt to fully automate compliance interpretation or legal analysis. Instead, it is designed as a quality-assurance tool that integrates into existing compliance workflows.

The software artifact has two identified use cases:

- To help companies prepare for an audit by collecting all company guidelines for each normative requirement.
- To help companies write better, more comprehensive guidelines, by checking which normative requirements lack coverage or has too many different guidelines.

5.3 Key Features

The following key features are at the core of *TraceAlign*:

- A feature for parsing and structuring normative requirements and company guidelines.
- A similarity search pipeline that proposes potential trace links using embedding-based search, NLI-based reranking, and LLM analysis.
- Storage of trace relationships in a Neo4j-based knowledge graph for querying and information extraction.
- Report generation for illustrating trace links from company guidelines to normative requirements and vice versa.

5.4 Design of the Software Artifact

Figure 5.1 shows the flowchart of the software artifact. It is designed as a web server with a REST API[1] interface and is built in Python.

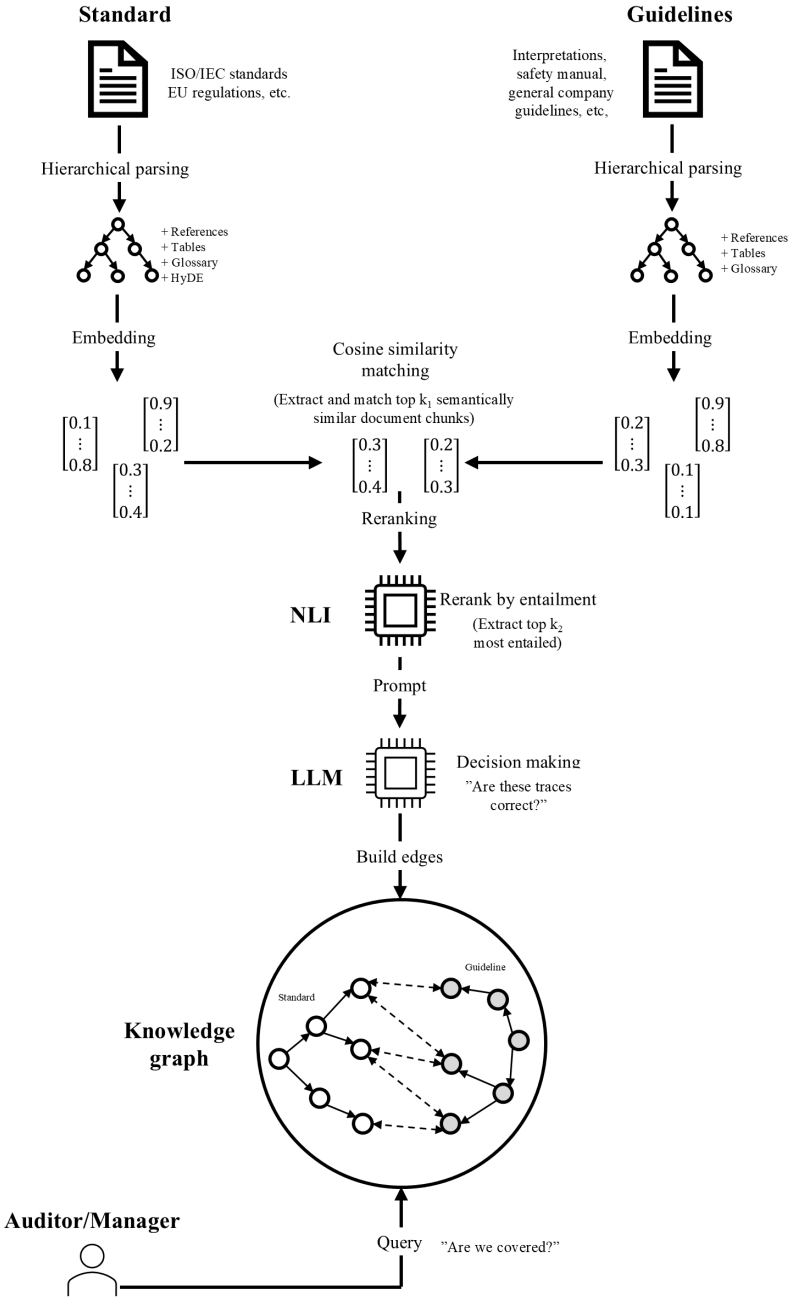

Figure 5.1: Flowchart of the software artifact.

[1]A REST API (Representational State Transfer API) is a web service that uses standard HTTP methods for stateless communication between clients and servers. The JSON format is most often used for simplicity and scalability [113].

5.4.1 Detailed Description of TraceAlign

This section describes in detail how *TraceAlign* works.

First, the user imports a .docx file with the standard and a .docx file with a company guideline that is to be traced. The inputted documents are parsed hierarchically in a deterministic way. Figure 5.2 shows the result of the document parsing on a conceptual level.

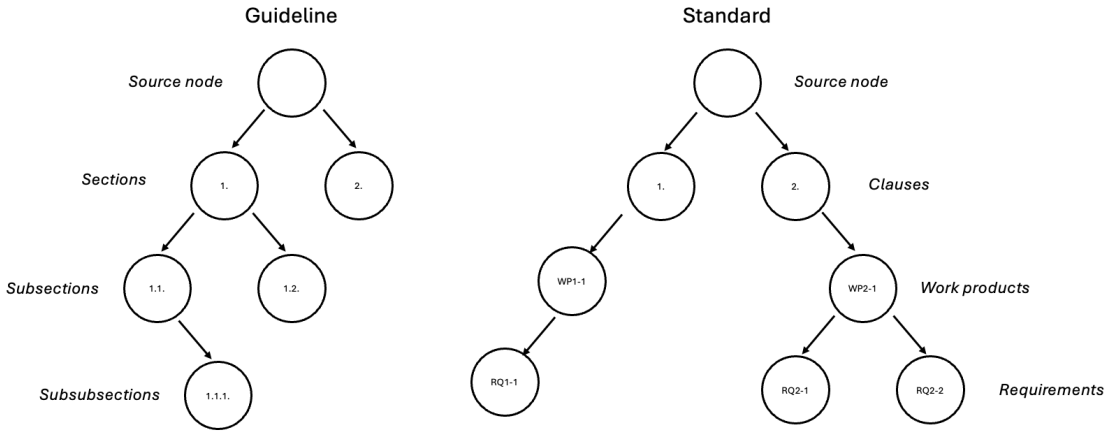

Figure 5.2: The result of document parsing in *TraceAlign*.

For the standard, the clauses, requirements, and work products are extracted and made into a tree-like structure in JSON where clauses point toward work products, work products point toward requirements, and requirements point toward subrequirements. For standards without explicit work products (like IEC 62443-4-1 [5]), dummy work products are built. For each of the extracted entities, their textual content is saved in their respective JSON object. Any referenced tables are converted into Markdown and also stored in the JSON object.[2] *TraceAlign* also generates glossary content based on a dictionary that has terms and explanations respectively. If a word is in the glossary and it is the first occurrence of that word, it gets a parenthesis with the explanation following the occurrence of the word. Listing 5.1 shows a redacted example of a requirement from ISO 21434 [6] represented as a JSON object.

The guidelines are divided into their respective sections with subsections for children. This is done all the way down to "heading 4" in the regular .docx heading formats. Like the standards, the textual content is saved in the JSON object, and tables are parsed into Markdown and then saved as well.

After the data are parsed, the resulting structured JSON objects are used for the creation of nodes and edges in the Neo4j database. To further illustrate the structure of the graph, an example with IEC 62443-4-1 [5] is shown in Figure 5.3. The purple

[2]The Markdown format is the most common way to provide tables to an LLM [114]. Tables that contain more than 500 tokens are summarized by an LLM.

```
"RQ-0X-0Y": {
        "content": "[RQ-0X-0Y] The (...) threat scenario (...) in
           accordance with [RQ-0Z-0Y]. ",
        "glossary_content": "[RQ-0X-0Y] The (...) threat scenario (
           Potential cause of compromise (...) a damage scenario)
           (...) in accordance with [RQ-0Z-0Y].",
        "tables": "",
        "children": {
            "RQ-0Z-0Y": {
                "content": "[RQ-0Z-0Y] The (...) cyberesecurity
                   (...).",
                "glossary_content": "[RQ-0Z-0Y] The (...)
                   cybersecurity (Condition in which assets are
                   sufficiently protected (...) components).",
                "tables": "",
                "children": {}
            }
        }
    }
```

Listing 5.1: Example JSON structure of a requirement from ISO 21434 [6].

node is the *StandardSourceModel*, the orange nodes are *ClauseModel*s, the light blue nodes are *WorkProductModel*s, and the dark green nodes are *RequirementModel*s.

All of the resulting nodes are then embedded into a 1024-dimensional vector space using BGE-M3 as the embedding model. For each of the resulting requirement nodes, *TraceAlign* also generates HyDE content that gets embedded in the same way as the regular content. The HyDE content is created by prompting an LLM with the content of the requirement node and asking it to rewrite it as a hypothetical company guideline. Table 5.1 shows all node types and their properties in Neo4j.

When the trace generation feature is invoked, *TraceAlign* starts with doing a cosine similarity search (shown as *Cosine similarity matching* in Figure 5.1). For each of the requirement nodes from the standard, their HyDE content is used to search for semantically similar guideline documents. The top k_1 retrieved guideline documents are then inputted to the NLI-reranker (nli-deberta-base) and get a new similarity score based on their respective entailment towards the glossary content of the requirement (see *NLI* in Figure 5.1). After reranking, the top k_2 results are kept.

Before generating the traces in Neo4j, the results from the reranking are given to an LLM for a more explicit reasoning step regarding whether the proposed traces are correct. The LLM is given the content of the normative requirements, all of its sub-requirements, referenced tables, and the contents of the top-k_2 possible traces. Based on this information, the LLM is prompted to output a confidence score from 0-3, representing how certain it is that each of these traces is correct, which is used as a filter. Based on the output from the LLM, the corresponding edges (traces) above a threshold are generated in Neo4j, and tracing is enabled (see *LLM* and *Knowledge graph* in Figure 5.1). An example of two complete traces from ISO 21434

Node Type	Property	Data Type
StandardSourceModel	uid	string
	source	string
	compliant	boolean
	date	date
ClauseModel	uid	string
	objectives	string
	embedding	vector
	section	string
	clause	string
WorkProductModel	uid	string
	tables	string
	content	string
	embedding	vector
	section	string
RequirementModel	uid	string
	tables	string
	hyde_content	string
	content	string
	gloss_content	string
	embedding	vector
	section	string
GuidelineSourceModel	uid	string
	source	string
	compliant	boolean
	date	date
SectionModel	uid	string
	tables	string
	hyde_content	string
	content	string
	name	string
	embedding	vector

Table 5.1: Node types and their properties in Neo4j

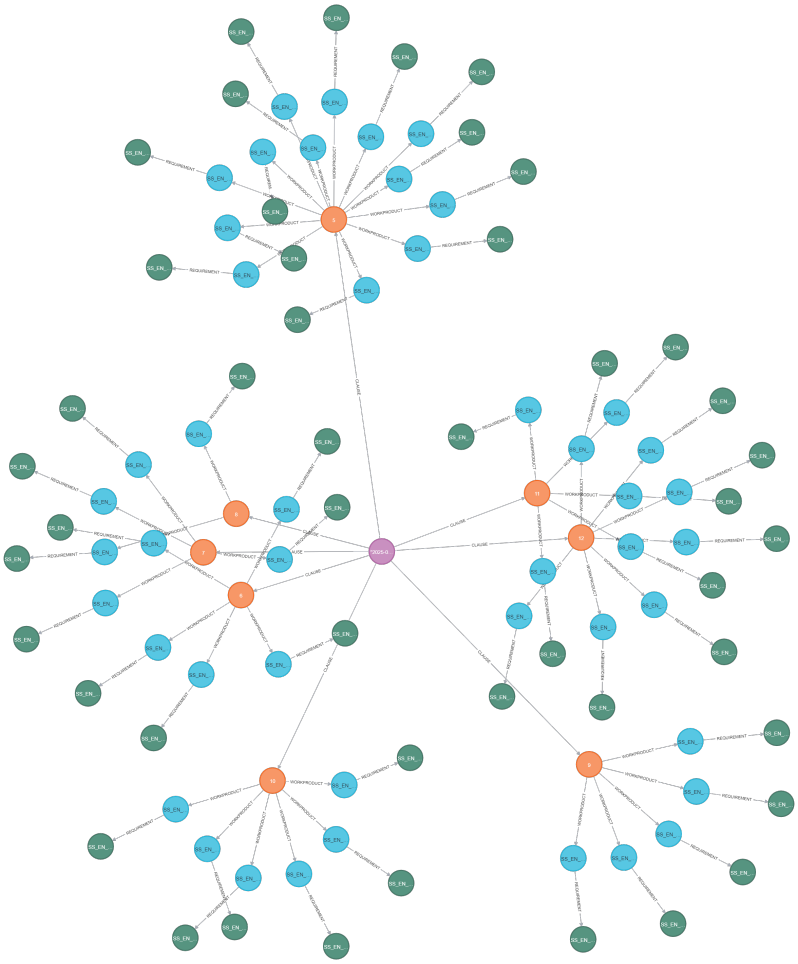

Figure 5.3: IEC 62443-4-1 represented in Neo4j.

[6] to R155 [115] generated by *TraceAlign* are shown in Figure 5.4. The purple node on the left is ISO 21434 [6] and the red node on the right is R155 [115]. Between the two "source nodes" the complete trace is shown (left to right): from clauses, to work products, to requirements, to sub-sections, and finally to a section.

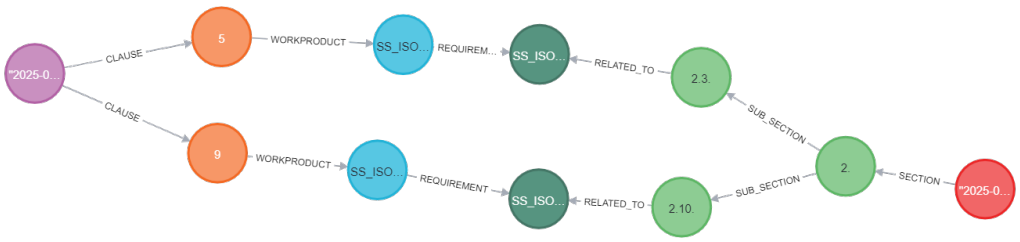

Figure 5.4: Two generated traces done by *TraceAlign*.

It should be noted that the generated traces are not guaranteed to be correct and should be reviewed by a domain expert.

The coverage report feature queries the Neo4j database for all requirements in one or more normative requirements (see *Query* from the Auditor/Manager in Figure

5.1). For each of the gathered requirements, the artifact checks if there are any traces present. If there exists a trace to a guideline, the traced guideline node is saved. With all of the saved guideline nodes and all of their respective requirements, a .docx report is generated containing the requirements as headings and a list of traced guideline sections as content (or vice versa). Figure 5.5 shows an example of a redacted coverage report generated based on ISO 21434 [6] and ISO 26262 [4].

5.5 Design Rationale

The design of *TraceAlign* was informed by previous works in NLP- and LLM-based automation of compliance (see Section 4.2). Notably, the decision to structure the normative requirements and guidelines as knowledge graphs was inspired by the works of Kruiper et al. [18], Elluri et al. [19], and Agarwal et al. [21]. Unlike previous works, which creates and populates a knowledge graph based on the semantic structures of the sentences, *TraceAlign* uses Regex-based pattern matching and the .docx format to recreate the structure of the document as a knowledge graph. This method appears to be novel in the compliance-automation field and was inspired by feedback from the focus group held at Company B during Cycle 2. Industry practitioners who have worked extensively with safety-critical standards highlighted that the current structure is how they would have represented normative requirements as a knowledge graph.

TraceAlign was particularly inspired by Agarwal et al. [21], which introduced the concept of storing crosswalks in a graph structure, where crosswalks are mappings between a source document and a set of target documents. This is identical to the task *TraceAlign* performs, where the normative requirements (e.g., ISO 26262, IEC 62443-4-1, or ISO 21434) are the source document and company guidelines are the target documents. However, unlike the work of Agarwal et al. [21] that is trained on evaluating all possible pairs of mappings between the documents using a BERT-model, *TraceAlign* is inspired by traditional works in the RAG-literature and uses embeddings and an NLI-reranker to reduce the amount of possible pairs to evaluate. Moreover, *TraceAlign* utilizes a zero-shot instruction-tuned GPT to evaluate the traces, requiring no training data.

The use of embeddings to find possible traces was inspired by Sai et al. [116], which similarly creates mappings between company guidelines and external requirements, but focuses instead on detecting possible deviations. Using embeddings to retrieve possible traces is moreover inspired by the general literature on RAG, see e.g. Lewis et al. [52]. Since the normative requirements come from a different domain than the company guidelines, *TraceAlign* uses HyDE to create the embeddings, inspired by Gao et al. [54]. The use of HyDE was moreover motivated by quantitative experiments, see Section 6.1.

Finally, the design decision to include a reranker was motivated by previous works in RAG, where it has been shown to increase accuracy over solely using embeddings [55]. Specifically using an NLI-model to rerank was motivated by Hua et al. [20],

Coverage Report – 2025-01-01

The following coverage report was created for the document(s) ISO 21434 and ISO 26262 on 2025-01-01, 00:00:00.

These requirements are covered to 90%.

The proceeding sections show the coverage.

ISO 21434

RQ-0X-01

[RQ-0X-01] The organization shall (...).

Guideline UID	Guideline Name	Guideline Content
[REDACTED]	[REDACTED]	[REDACTED]

RQ-0X-02

No coverage.

RQ-0X-03

[RQ-0X-03] A plan shall be (...).

Guideline UID	Guideline Name	Guideline Content
[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]

RQ-0Y-01

RQ-0Y-02

RQ-0Z-01

ISO 26262-6

5.4.1.

5.4.2.

6.4.1.

7.4.1.

Figure 5.5: Example of a coverage report generated by *TraceAlign*.

which illustrated that such models could, to a certain extent, interpret both company policies and regulatory texts to meaningfully infer whether they are contradictory. The use of an NLI-reranker was moreover shown to have a significant impact on the F_1-score of the artifact, compared to using only an embedding model or a traditional reranker (see Section 6.1).

5.6 Summary

TraceAlign automates the task of compliance tracing of company guidelines to normative requirements. It uses Regex-matching and headings to create hierarchical knowledge graph-representations of both normative requirements and company guidelines. Further, it utilizes word embeddings, an NLI model, and an LLM to create traces between the documents. The development of *TraceAlign* is the final outcome of the iterative design process carried out in this study. The next chapter presents the evaluation of the software artifact (*RQ3*), where its effectiveness and usability were assessed through quantitative experiments and practitioner feedback.

6

Evaluation

This chapter answers *RQ3*: *To what extent can an AI-system, relying on LLM-technology, sufficiently automate part of the compliance process in a firm?* The answer for *RQ3* is based on the results from the quantitative factorial experiments and focus groups done in Cycle 3.

6.1 Factorial Experiments

The software artifact was evaluated on its ability to trace between a regulation and a standard, using data manually labeled by domain experts. Originally, this dataset was mainly used in the design and development of the software artifact, however, it also helps answer *RQ3* as it quantitatively measures how well the artifact performs the task. It is important to note that the dataset is heavily imbalanced since the number of true mappings is very small compared to the number of possible mappings. As such, accuracy is a poor metric for capturing how well the software artifact is performing. For reference, if the artifact predicts that there should be no mappings between the regulation and standard, it will achieve an accuracy of almost 90%, despite being practically useless. As such, accuracy will be accompanied by the metrics recall, specificity, and F_β-score.

The software artifact consisted of three interconnected components: the embedding model, the reranking model, and the LLM. Each one of these models can be exchanged, and for each model there are a number of parameters that can be adjusted. As such, a full factorial experiment testing every possible combination for the pipeline is infeasible. Instead, factorial experiments were conducted component-by-component and then a few final experiments compared a subset of all factors for the entire pipeline.

6.1.1 Embedding Model

Five different embedding models were experimented with: Instructor-XL, BGE-M3, LegalBert [117], a BERT-model fine-tuned on the standard, and a BERT-model fine-tuned on both the standard and the regulation.[1]

[1]Specifically, the Bert-model 'bert-base-uncased' was trained for 10 epochs with an AdamW optimizer from Pytorch with a learning rate of $5e^{-5}$ on a masked language modeling task for these

What content/input should be embedded is also a qualitative parameter that can be adjusted. For this, the authors experimented with four different variations to the embeddings: no change (i.e., the text exactly as written in the standard and guidelines are embedded), cleaned (i.e,. stop words were removed), glossary (i.e., definitions for words were added), and HyDE.

Finally, for each of these embedding models and inputs, the top k_1 passages retrieved will be passed down to the next stage in the pipeline. What k_1 should be and how it should be computed are also parameters that can be adjusted. The authors experimented with three different types of thresholds: normalized similarity scores,[2] Z-score,[3] and simply picking the top k_1 answers.

Thus, a total of 20 experiments were carried out, one for each embedding model and input. All of these were then evaluated based on the different thresholds; however, computing these different thresholds did not require running the experiments again, and are thus not treated as a factor, but rather different ways to evaluate the result. The full outcome of the experiments can be found in Appendix E.1.

Generally, BGE-M3 performed the best and, as can be seen in Figure 6.1, HyDE generally outperformed the other variations to the embeddings. For example, at $k_1 = 6$ HyDE gives a recall of 79% and an F_1-score of 36%. No processing of the input, i.e., normal, gives a recall of 66% with an F_1-score of 30% for the same k_1.

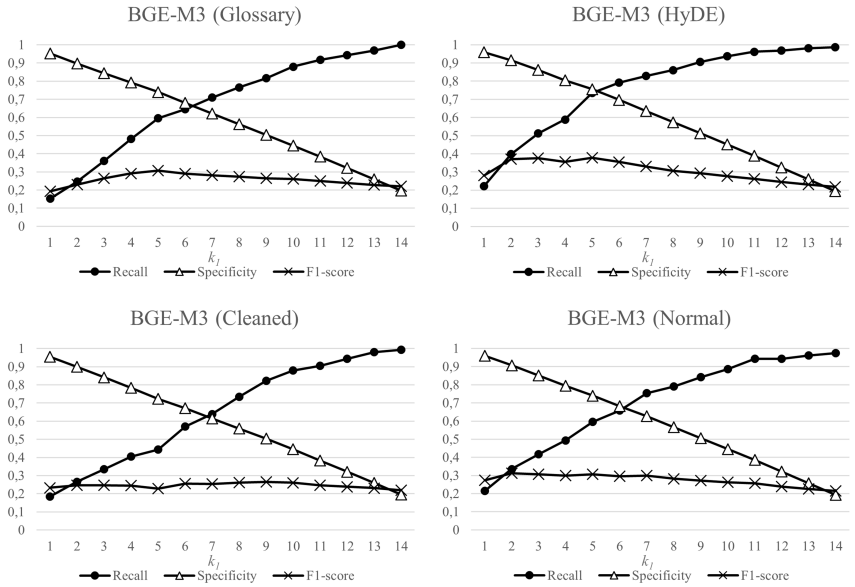

Figure 6.1: Outcomes of different input texts on BGE-M3.

documents, where 15% of the words were masked at random.

[2]The method used was min-max normalization. In other words, given a list of numbers $\mathbf{x} = [x_1, ..., x_n]$ a new list $\mathbf{x}' = [x'_1, ..., x'_n]$ was created where $x'_i = \frac{x_i - \max(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}$ $\forall i = 1, ..., n$.

[3]The method used for computing Z-score (also known as standard score) was, given a list of numbers $\mathbf{x} = [x_1, ..., x_n]$ create a new list $\mathbf{x}' = [x'_1, ..., x'_n]$ where $x'_i = \frac{x_i - \mu}{\sigma}$ $\forall i = 1, ..., n$ and where μ is the mean of the list and σ is the standard deviation of the list.

6.1.2 Reranking Model

Three different rerankers were experimented with: bge-reranker-v2-m3 [118], [119], qnli-electra-base [120], and nli-deberta-base [121]. bge-reranker-v2-m3 and qnli-electra-base were trained on determining whether a given passage can answer a given question, while nli-deberta-base was trained on determining whether two passages are "entailed", "contradictory" or "neutral". The same threshold variants as used for the embedding model were experimented with here as well. Note that for the nli-deberta-base model, passages were ranked according to how high entailment scores they were given.[4]

The full results of the experiments can be found in Appendix E.2. All experiments were conducted on text with glossary definitions. As such, only one factor was varied, i.e., which model is used. Figure 6.2 shows the difference in performance between the different models. Evidently, nli-deberta-base outperforms the two question-answer rerankers, strengthening the hypothesis that the two traced sections should be entailed. For reference, nli-deberta-base achieves a recall of 70% and F_1-score of 43% at $k_2 = 4$. qnli-electra-base and bge-reranker-v2-m3 only achieves recalls of 30% and 28% and F_1-scores of 18% and 17% for the same k_2, respectively.

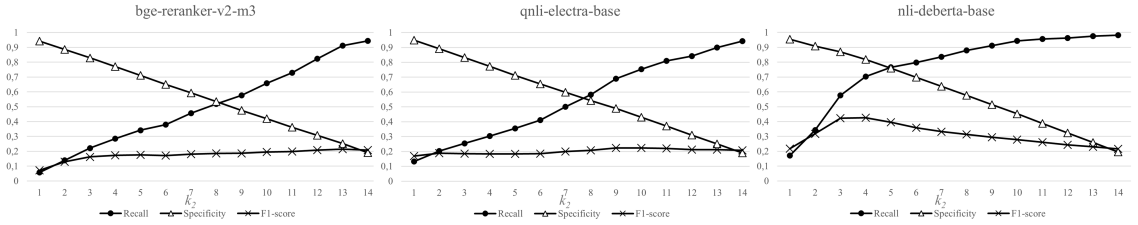

Figure 6.2: Outcomes of different reranker models.

6.1.3 Large Language Model

The outcome of the LLM component of the pipeline is heavily dependent on two factors: which LLM is used and what prompt is given.

The LLM was varied between two models of the same size: Llama-3.3-70b[5] [122] and Deepseek-r1-70b[6] [123]. The former is a traditional instruction-tuned GPT and the latter is a "reasoning" model, i.e., a model that has been trained to include a reason for the way it chooses to follow instructions. The prompt was varied between asking the model to return a confidence score between 0-3 for each trace and asking it to also return a reason for each confidence score. The "reason"-prompt has a dual purpose, since forcing the model to include a reason can improve the performance [124], and this information could also be useful for the end user.

[4] Alternative set-ups were also briefly experimented with, such as ranking based on the negative contradiction score, and simply using it as a classifier where only passages that were "entailed" were included. These setups appeared to perform slightly worse.

[5] Version ID a6eb4748fd29 in ollama.

[6] Version ID 0c1615a8ca32 in ollama.

The full results of the experiments can be found in Appendix E.3. All experiments were conducted based on the original text given in both the standard and the regulation. The outcome of the experiments is illustrated in Figure 6.3. As can be seen, forcing the LLM to generate a reason generally increases recall but decreases specificity. Overall, it has a small positive effect on the F_1-score. There are clear differences in the behavior of the different LLMs, however, it is unclear which performs the best.

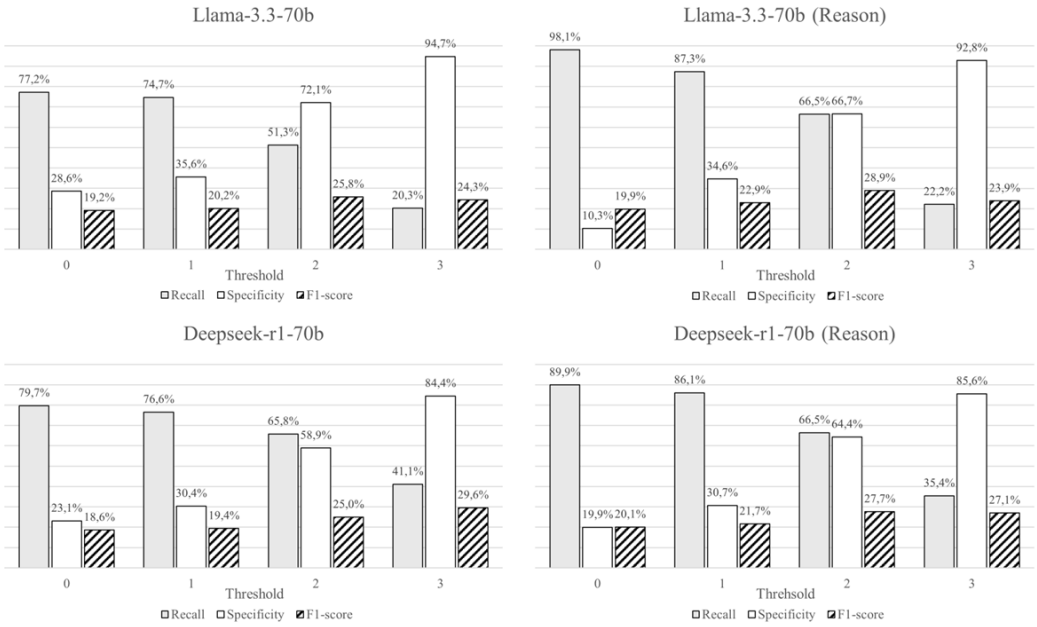

Figure 6.3: Outcomes of different LLMs and prompts.

6.1.4 Final Experiments

Finally, a subset of all possible factors was experimented on to evaluate the full software artifact and determine the effects of varying them. The following three factors were chosen: using HyDE vs. using normal for the embeddings; using nli-deberta-base to rerank vs. using no reranker; and forcing the LLM to output a reason vs. not forcing it to output a reason. All experiments were run using the BGE-M3 embedding model, Llama-3.3-70b with its threshold set to 1 as the LLM, $k_1 := 15$, and $k_2 := 5$.

All metrics from the experiments can be found in Appendix E.4. Figure 6.4 shows the outcome of the experiments based on recall, specificity, and F_1-score. As can be seen, the NLI-reranker had the largest impact on performance, since experiments including it significantly increase the recall. Judging by the F_1-score, it would appear that the best combination is using HyDE, an NLI-reranker, and not forcing the LLM to reason. This results in a software artifact able to make 73% of traces that should exist while removing 82% of the traces that should not exist.

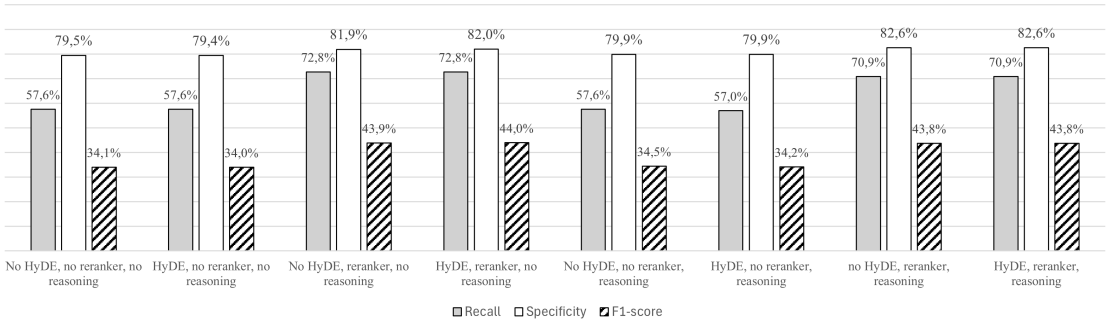

Figure 6.4: Outcome of the full experiments.

6.2 Focus Group

This section describes the results of the thematic analysis done for two focus groups held at two different companies, in different domains, with different levels of compliance maturity. For the focus group held at Company B, the standard *TraceAlign* was evaluated on was ISO 26262, and for the focus group held at Company D, the standard was IEC-62443-4-1.

In general, both focus groups identified that using *TraceAlign* would be time- and cost-reducing compared to the current manual way of doing compliance tracing. This result was highlighted by participants several times: *"I see it as you have proved kind of the idea that it actually works and, and could provide some value for companies"* and: *"yes, I think it's useful and it could be time-saving"*. However, one focus group did not believe that *TraceAlign* would help increase the accuracy of their tracing, as the software artifact would occasionally make incorrect traces that a human would not make. This was noted as: *"I don't think it will make our traces more accurate, but in much shorter time"*.

Many of the participants however argued that even these incorrect traces were oftentimes useful. They would sometimes point to the wrong guideline, but that guideline would then reference the correct guideline. One participant mentions: *"It at least guides you in the right direction"*, which highlights the usefulness. Several participants highlight that the incorrect trace could also signal to the user that the correct guideline did not exist or was confusingly written: *"you get kind of a hint of, OK, do we miss anything?"*, and: *"if we split fulfillment of one requirement into multiple guidelines, maybe it's confusing for the end user"*. For example, the AI made one trace where a requirement was split over two different sections. This led the compliance practitioners to question why the requirement was split and pointed out that this guideline was perhaps confusing for the user.

In general, the participants expressed that this was a key purpose and use of *TraceAlign*, i.e., to help them write better guidelines and processes: *"I think maybe it can also help structuring of guidelines"*, *"(...) could be an instrument to improve"*. It was reasoned that if the AI made incorrect traces or could not find the right guide-

line, then likely there was something off about that guideline. As such, running the software artifact occasionally on all guidelines and normative requirements would help identify gaps and errors in the current processes and help with continuous improvement and quality control. This was highlighted by one participant as: *"(...) ongoing improvement where we can run this system, maybe I don't know, every second month or something to see progress and filling the gaps, lowering the the number of red dots in the heat map and, and things like that, that would be really useful"*. Moreover, if *TraceAlign* then identifies major areas where the company lacks guidelines, then it could also help with planning and resource allocation, as this would require significant effort to fix. This was noted as: *"If we identify a huge area where we are weak, then we need to assign some resources to fix it"*.

Another automation use case that the practitioners identified was to use the software artifact to prepare for audits. Often when the company prepares for an audit, it must produce a compliance matrix showing for each normative requirement which processes and guidelines support its fulfillment. This is tedious and time-consuming, and could be automated with the use of *TraceAlign*. This was highlighted in one of the focus groups: *"when we have audits, to make this matrix, you could say, for this set of requirements, look for evidence over here"*. Since *TraceAlign* runs much quicker than doing it manually, this would save significant time.

Finally, both focus groups expressed that it was clear what part of the compliance process the *TraceAlign* was trying to automate and at a high level how it did it.

6.3 Summary

The quantitative experiments performed showed that the software artifact struggled to reach a high F_1-score. However, these results are in a different, although conceptually related domain, and are not representative of actual performance. They show that the software artifact traces well, but not perfectly, which agrees with the consensus of the focus groups. However, errors in traces can come from many different sources, both from the companies' guidelines and mistakes in *TraceAlign*. These mistakes are of value and can illustrate gaps in the company's guidelines and/or show that they are written in an unclear manner. Figure 6.5 summarizes key feedback from the focus groups divided into strengths and weaknesses.

Strengths				**Weaknesses**
Value-creating		**Multi-use**		Inaccurate
Reduces time	Reduces cost	Audit preparation	Guideline improvement	

Figure 6.5: Strengths and weaknesses of *TraceAlign*.

With regards to *RQ3*, an AI system leveraging LLM technology can automate compliance tracing to a meaningful extent. While it may not fully replace human expertise, the software artifact can support the tracing of normative requirements to

company guidelines. This level of automation can reduce manual effort, enhance audit preparedness, and facilitate continuous improvement of compliance-related processes and guidelines.

7

Discussion

This section summarizes and discusses the findings from Chapters 4 and 6. The discussion has been divided into three sections: implications for research, implications for practice, and validity and ethical considerations.

7.1 Implications for Research

The outcome of the interviews, literature review, and focus groups and their subsequent consequences on the design and utility of *TraceAlign* have implications for the general research surrounding the challenges of compliance and compliance automation.

Several challenges related to compliance in software-intensive firms were identified based on the interviews. Many of these challenges were high-level, organizational and management-related problems that are difficult to solve using an NLP/LLM-based software artifact. Nevertheless, they present important challenges that need to be solved. These challenges are perhaps better approached through a management perspective, using for example the lens of business process re-engineering or total quality management. Some of the challenges identified were however fine-grained and can be addressed using NLP/LLM methods, and many of them already have been, as evidenced by the literature review. These results contribute to the previous knowledge on what challenges software-intensive companies face with compliance and introduce some avenues for future research in the field of NLP/LLM-enabled compliance automation.

Based on the interviews and literature review, the authors identified a research gap with regard to automatic tracing from company guidelines to normative requirements. This challenge was the key focus of the thesis and the developed software artifact, *TraceAlign*, was an attempt to fill this research gap. Tracing from company guidelines to normative requirements is a field in which very little prior research has been published. The literature review identified only one relevant paper, Sai et al. [116], which focused on detecting deviations between company guidelines and normative requirements. Sai et al. [116] also present a general approach which is illustrated on both a standard and a regulation, however, the method presented relies heavily on manual inspection and traditional NLP methods. *TraceAlign* is

however fully automatic and uses an instruction-tuned GPT for decision-making, although results could be manually inspected to improve performance. Agarwal et al. [21] introduced a related system for creating crosswalks in general; however, unlike *TraceAlign*, their solution relies on a fine-tuned BERT model and exhaustive pair-wise comparisons to create mappings. *TraceAlign* performs a similar task but requires no training data. Regardless, it is evident that compliance tracing is a field of great value to industry practitioners, which has received little attention from the scholarly literature.

From a technical view, *TraceAlign* was influenced by and extended several previous works in compliance automation. The idea of using a knowledge graph to represent the normative requirements was loosely inspired by the works of Kruiper et al. [18] and Elluri et al. [19]. The way *TraceAlign* builds the knowledge graph is however more closely related to the reference resolution introduced by Rahmani et al. [83] and the structure of the graph was informed by close collaboration with compliance practitioners. Hua et al. [20] moreover inspired the use of NLI models as a reranker. Conceptually, *TraceAlign* was heavily influenced by the crosswalking concept as characterized by Tupsumadre et al. [37] and the deviation detection between company policies and external regulations in Sai et al. [116]. *TraceAlign* extends both of these works by incorporating many of the latest developments in NLP, LLMs, and RAG to solve the identified challenge.

As for the three components of the RAG-pipeline, *TraceAlign* extends and strengthens some previous results. First of all, *TraceAlign* is further evidence that HyDE can have a positive effect on the performance of the retriever in RAG systems. Second, the use of an NLI model to rerank the result from the retriever appears to have been a novel idea in the domain of compliance tracing. It is conceptually related to the work of Hua et al. [20], however, *TraceAlign* uses entailment as a measure of the likelihood of a trace existing, as opposed to using contradiction as a measure of the likelihood of a deviation existing. The thesis was however unable to significantly improve the performance of the LLM-component, despite using different models and prompting techniques. Further work focusing on this component using larger LLMs and different prompting techniques could increase the performance.

Finally, the DSR paradigm was a central piece of the field study presented in this thesis. Since the thesis was conducted in close collaboration with industry, DSR provided a framework that allowed the researchers to produce a software artifact that was not only novel but also valuable to at least some of the partnering companies. That DSR would provide a framework that resulted in a valuable software artifact is however unsurprising considering the many previous success stories [61], [62]. Nevertheless, this thesis contributes to the general knowledge on which type of challenges are well-suited for DSR.

7.2 Implications for Practice

The main implication for practice is that there are many different challenges with compliance and many different opportunities to solve these challenges using NLP and LLMs. This thesis choose to focus exclusively on solving the small but significant challenge of compliance tracing. Despite the narrow focus, many focus group participants remarked that the software artifact could have several implications for the companies' compliance processes. Importantly, two main use cases were highlighted by both Cycle 3 focus groups:

First, being able to produce traces from all normative requirements that a company must comply with to all of their guidelines could be time- and thus cost-saving in preparing for an audit. As highlighted by the industry practitioners, audits occur often and require extensive manual labor currently. Automating this task at a fraction of the time and cost could have major implications for the competitive advantage of the companies.

Second, both focus groups highlighted that the proposed software artifact could help them write better guidelines. By running the artifact occasionally on all normative requirements and guidelines, companies can identify missing or confusing guidelines that have not been traced by the artifact. Thus, *TraceAlign* has a dual purpose as a quality assurance tool which in turn could support continuous improvement and other compliance-related work within the company.

All in all, the thesis has highlighted many challenges and automation opportunities and attempted to solve only a small subset of them. Much work remains before the compliance processes in firms can be fully automated, if ever. If anything, it can be concluded that all software-intensive companies interviewed face costly and time-consuming challenges with compliance. Any tool which can partially automate any one of these challenges are surely welcomed by these companies.

7.3 Validity and Ethical Considerations

This section covers some considerations regarding the internal and external threats to the validity of the thesis, as well as some ethical considerations. Regarding the generalizability of the results, there are a number of considerations of note.

First of all, the interviewees were all representatives of European software-intensive companies. The regulatory environment of companies is heavily dependent on their industry and geographical location, and as such nothing would suggest that the findings generalize outside of Europe and software-intensive companies. Therefore, no such claim is made. However, the interviewees mostly possessed different roles and worked with different normative requirements in different industries, strengthening the internal validity of the results.

The data from the interviews and subsequent analysis of said data is also to subjected

to some threats to internal validity. Regarding the interviews, some potential threats are the interviewers' initial lack of knowledge regarding the subject matter and the potential sensitive nature of the questions asked. To mitigate the researchers' lack of knowledge regarding compliance, semi-structured interviews were utilized to allow interviewees and interviewers to steer the questions and answers dynamically during the interview. To avoid the interviewees not disclosing sensitive information, non-disclosure agreements were signed. Regarding the analysis, any qualitative data analysis is subject to some researcher bias and would likely not turn out exactly the same if repeated. To mitigate this bias, all thematic analyses were always performed by both researchers.

The software artifact was moreover tested and evaluated on ISO 21434, ISO 26262, and IEC 62443-4-1, three safety-critical process standards. While the multitude of standards strengthens the internal validity of the results, their narrow scope and domain limit the generalizability to normative requirements in general. Applying, adapting, and evaluating *TraceAlign* on different normative requirements in different companies is therefore left as future work.

The qualitative focus group evaluation of the final software artifact presents some threats to internal validity that the researchers have attempted to mitigate. First of all, the artifact is displayed in an artificial setting with the researchers controlling the tool. This is of course not representative of a real use case for *TraceAlign*, however, teaching each participant how to use the tool and allowing them to test it would be too time-consuming. To mitigate this discrepancy, real company guidelines and standards were used in the demonstration sessions, and the participants were encouraged to instruct the researchers on which requirements and guidelines were interesting to see. Further, the fact that the software artifact was evaluated in two different companies with different guidelines and standards, yet the results were largely the same, strengthens the internal validity.

The quantitative factorial experiments on a labeled dataset do little to support the internal validity of the study, however. The experiments were conducted on ISO 21434, a different safety-critical process standard than the ones evaluated qualitatively in the focus groups. As such, it cannot be ascertained whether the software artifact's performance on this standard is necessarily representative of its performance on the other standards. Moreover, and perhaps more importantly, the quantitative results show the artifact's performance on a different task. While arguably there is some conceptual resemblance between tracing from a regulation to a standard and tracing from a standard to company guidelines, nothing can be said about whether these tasks are similar in practice or of equal difficulty. As such, all quantitative results presented in the study should be interpreted with much care. They were not included to accurately depict the software artifact's true performance, but rather to show how the artifact was designed and different parameters were tuned. To a certain degree, they also capture in some quantitative sense how close *TraceAlign* comes to capturing "human intuition".

7.3.1 Ethical Considerations

The main ethical consideration of the thesis is related to informed consent regarding the interviewees and focus group participants. All interviewees and focus group participants were informed that the results would be used to write this master's thesis and that the audio would be recorded and transcribed but not disseminated. All participants were moreover informed that all data would be anonymous in the final thesis.

Finally, the authors are bound by a non-disclosure agreement with all participating companies to not disclose any trade secrets. As such, specifics regarding the companies' guidelines and interpretations of standards cannot be disclosed in this thesis. Moreover, the dataset used for the quantitative factorial experiments can regrettably not be shared.

8

Conclusion

Using the DSR approach, this thesis has: 1) identified several compliance challenges for software-intensive companies; 2) identified several automation opportunities in relation to these challenges and; 3) created an NLP- and LLM-enabled software artifact to partially automate compliance tracing between company guidelines and normative requirements.

Based on semi-structured interviews with four European software-intensive companies, several compliance challenges could be identified. The identified challenges were divided into five themes, of which this thesis focused specifically on "Challenges Related to Requirements", particularly the challenge of tracing company guidelines to normative requirements. This challenge was common to all software-intensive companies and no prior work was found that addressed the challenge. In close collaboration with two of the aforementioned firms, this thesis designed and developed an NLP- and LLM-based software artifact to automate the challenge.

The key contribution of the thesis is the software artifact, *TraceAlign*. *TraceAlign*, is a RAG system which utilizes embeddings, NLI models, LLMs and a knowledge graph to trace between company guidelines and normative requirements and persist the results. When evaluated in focus groups with the participating companies, *TraceAlign* was believed to be time- and cost-saving for the companies although likely not accuracy-increasing compared to the traditional manual way of performing the task. Importantly, two use cases were identified:

- *TraceAlign* could be used to produce information in preparation for an audit, a traditionally time-consuming and tedious task.
- *TraceAlign* could be used to check the coverage of the company's guidelines, ensuring that the company works with continuous improvement and ensuring that no guidelines are missing or incorrectly written.

TraceAlign is a first attempt to automate compliance tracing between company guidelines and normative requirements. Much future work remains, particularly in increasing its accuracy and evaluating it on other normative requirements in different domains. There is moreover an abundance of other challenges and automation opportunities identified in the thesis, many of which could perhaps be solved using

the latest developments in NLP and LLMs.

Bibliography

[1] Jan vom Brocke, Alan Hevner, and Alexander Maedche. *Introduction to Design Science Research*, pages 1–13. Springer International Publishing, Cham, 2020.

[2] Cambridge University Press. Compliance. https://dictionary.cambridge.org/dictionary/english/compliance, 2025. Accessed: 2025-03-26.

[3] Veronica Root. The compliance process. *Indiana Law Journal*, 94(1):5, 2019.

[4] International Organization for Standardization. ISO 26262: Road Vehicles – Functional Safety. https://www.iso.org/standard/68383.html, 2018. Second edition, ISO 26262 series (Parts 1–12).

[5] International Electrotechnical Commission. IEC 62443 Series — Industrial communication networks – IT security for networks and systems. Standard, 2010. Available from: https://webstore.iec.ch/publication/series/614.

[6] International Organization for Standardization and SAE International. ISO/SAE 21434:2021 Road Vehicles — Cybersecurity Engineering. Standard, 2021. Available from: https://www.iso.org/standard/70918.html.

[7] European Commission. Cyber resilience act. https://digital-strategy.ec.europa.eu/en/policies/cyber-resilience-act, March 2025. Last updated on 6 March 2025.

[8] European Parliament and Council of the European Union. Regulation (eu) 2016/679 of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (general data protection regulation). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2016.119.01.0001.01.ENG, May 2016. OJ L 119, 4.5.2016, pp. 1–88.

[9] Jane Claydon. *Compliance/Legal Compliance*, pages 429–434. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[10] Statista. Largest fines issued for general data protection regulation (gdpr) vio-

lations as of february 2025 (in million euros) [graph]. https://www.statista.com/statistics/1133337/largest-fines-issued-gdpr/, 2025. GDPR Enforcement Tracker, February 17, 2025. [Online].

[11] European Parliament and Council of the European Union. Regulation (eu) 2023/2854 of the european parliament and of the council of 13 december 2023 on harmonised rules on fair access to and use of data and amending regulation (eu) 2017/2394 and directive (eu) 2020/1828 (data act), 2023.

[12] European Parliament and Council of the European Union. Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence and amending regulations (ec) no 300/2008, (eu) no 167/2013, (eu) no 168/2013, (eu) 2018/858, (eu) 2018/1139 and (eu) 2019/2144 and directives 2014/90/eu, (eu) 2016/797 and (eu) 2020/1828 (artificial intelligence act), 2023.

[13] Shannon W Anderson, J Daniel Daly, and Marilyn F Johnson. Why firms seek iso 9000 certification: Regulatory compliance or competitive advantage? *Prod. Oper. Manag.*, 8(1):28–43, March 1999.

[14] Zachary Garner. What do UNR 155 and ISO/SAE 21434 mean for OEMs? - WirelessCar — wirelesscar.com. https://www.wirelesscar.com/what-do-unr-155-and-iso-sae-21434-mean-for-oems/. [Accessed 23-04-2025].

[15] Julieth Patricia Castellanos Ardila, Barbara Gallina, and Faiz Ul Muram. Compliance checking of software processes: A systematic literature review. *Journal of Software: Evolution and Process*, 34(5):e2440, 2022.

[16] Francesco Trebbi and Miao Ben Zhang. The cost of regulatory compliance in the united states. Technical report, Cambridge, MA, November 2022.

[17] Muhammad Usman, Michael Felderer, Michael Unterkalmsteiner, Eriks Klotins, Daniel Mendez, and Emil Alégroth. Compliance requirements in large-scale software development: An industrial case study. In Maurizio Morisio, Marco Torchiano, and Andreas Jedlitschka, editors, *Product-Focused Software Process Improvement*, pages 385–401, Cham, 2020. Springer International Publishing.

[18] Ruben Kruiper, Bimal Kumar, Richard Watson, Farhad Sadeghineko, Alasdair Gray, and Ioannis Konstas. A platform-based natural language processing-driven strategy for digitalising regulatory compliance processes for the built environment. *Adv. Eng. Inform.*, 62(102653):102653, October 2024.

[19] Lavanya Elluri, Sai Sree Laya Chukkapalli, Karuna Pande Joshi, Tim Finin, and Anupam Joshi. A bert based approach to measure web services policies compliance with gdpr. *IEEE Access*, 9:148004–148016, 2021.

[20] Min Hua, Qi Zhao, Jiale Song, and Xue-song Tang. Two-stage compliance de-

tection for power enterprises based on nli and llm. In *2024 IEEE International Symposium on Product Compliance Engineering - Asia (ISPCE-ASIA)*, pages 1–5, 2024.

[21] Vikas Agarwal, Roy Bar-Haim, Lilach Eden, Nisha Gupta, Yoav Kantor, and Arun Kumar. Ai-assisted security controls mapping for clouds built for regulated workloads. In *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*, pages 136–146, 2021.

[22] Piërre van de Laar, Sjir van Loo, Gerrit Muller, Teade Punter, David Watts, Pierre America, and Joland Rutgers. The darwin project: Evolvability of software-intensive systems. In *Third International IEEE Workshop on Software Evolvability 2007*, pages 48–53, 2007.

[23] Derya Fındık and Aysit Tansel. Intangible investment and technical efficiency. In *Handbook of Research on Cultural and Economic Impacts of the Information Society*, pages 179–202. IGI Global, 2015.

[24] Jan Bosch and Helena H. Olsson. Digital for real: A multicase study on the digital transformation of companies in the embedded systems domain. *Journal of Software: Evolution and Process*, 33(5):e2333, 2021. e2333 JSME-20-0122.R2.

[25] Lee G Branstetter, Matej Drev, and Namho Kwon. Get with the program: Software-driven innovation in traditional manufacturing. *Management Science*, 65(2):541–558, 2019.

[26] Joseph J. Cordes, Susan E. Dudley, and Layvon Washington. Regulatory compliance burdens: Literature review and synthesis. Technical report, Regulatory Studies Center, Trachtenberg School of Public Policy & Public Administration, The George Washington University, October 2022.

[27] David Parker and Colin Kirkpatrick. The economic impact of regulatory policy: A literature review of quantitative evidence. Oecd expert paper no. 3, Organisation for Economic Co-operation and Development (OECD), August 2012.

[28] 2023 cost of compliance: Regulatory burden poses operational challenges for compliance. Technical report, Thomson Reuters, 2023.

[29] PwC. A privacy reset — from compliance to trust-building. `https://www.pwc.com/us/en/services/consulting/cybersecurity-risk-regulatory/library/privacy-reset.html`, 2021.

[30] Pragyan K, Rambod Ghandiparsi, Rocky Slavin, Sepideh Ghanavati, Travis Breaux, and Mitra Bokaei Hosseini. Toward regulatory compliance: A few-shot learning approach to extract processing activities. In *2024 IEEE 32nd International Requirements Engineering Conference Workshops (REW)*, vol-

ume abs/2104.04030, pages 241–250. IEEE, June 2024.

[31] Muhammad Ilyas Azeem and Sallam Abualhaija. A multi-solution study on GDPR AI-enabled completeness checking of DPAs. *Empir. Softw. Eng.*, 29(4), July 2024.

[32] Vad är en standard? - Svenska institutet för standarder, SIS — sis.se. `https://www.sis.se/standarder/vad-ar-en-standard/`. [Accessed 10-03-2025].

[33] ISO - Benefits of ISO standards — iso.org. `https://www.iso.org/benefits-of-standards.html`. [Accessed 10-03-2025].

[34] Michelle Egan. Setting standards: Strategic advantages in international trade. *Business Strategy Review*, 13(1):51–64, 2002.

[35] Food and Agriculture Organization of the United Nations. *Environmental and social standards,certification and labelling for cash crops*. FAO commodities and trade technical papers. Food & Agriculture Organization of the United Nations (FAO), Rome, Italy, May 2004.

[36] Luiz Eduardo G. Martins and Tony Gorschek. Requirements engineering for safety-critical systems: A systematic literature review. *Information and Software Technology*, 75:71–89, 2016.

[37] Harshal Tupsamudre, Arun Kumar, Vikas Agarwal, Nisha Gupta, and Sneha Mondal. Ai-assisted controls change management for cybersecurity in the cloud. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12629–12635, Jun. 2022.

[38] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd edition, 2025. Online manuscript released January 12, 2025.

[39] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[40] Djoerd Hiemstra. *Language Models*, pages 1591–1594. Springer US, Boston, MA, 2009.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you

need, 2023.

[42] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[43] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[44] Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, and Thomas B. Schön. *Machine Learning - A First Course for Engineers and Scientists*. Cambridge University Press, 2022.

[45] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[46] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[47] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[48] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[49] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[50] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022.

[51] Tom Taulli. *Prompt Engineering*, pages 51–64. Apress, Berkeley, CA, 2023.

[52] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[53] Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. Rq-rag: Learning to refine queries for retrieval augmented generation, 2024.

[54] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels, 2022.

[55] Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Ra-

jaram Naik, Pengshan Cai, and Alfio Gliozzo. Re2g: Retrieve, rerank, generate, 2022.

[56] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[57] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction, 2022.

[58] Juan Sequeda and Ora Lassila. *Introduction*, pages 1–17. Springer International Publishing, Cham, 2021.

[59] Juan Sequeda and Ora Lassila. *Designing Enterprise Knowledge Graphs*, pages 19–44. Springer International Publishing, Cham, 2021.

[60] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee and. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77, 2007.

[61] Stefan Seidel, Jan Recker, and Jan vom Brocke. Sensemaking and sustainable practicing: Functional affordances of information systems in green transformations. *MIS Quarterly*, 37(4):1275–1299, 2013.

[62] Jan vom Brocke and Stefan Seidel. Environmental sustainability in design science research: Direct and indirect effects of design artifacts. In Ken Peffers, Marcus Rothenberger, and Bill Kuechler, editors, *Design Science Research in Information Systems. Advances in Theory and Practice*, pages 294–308, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[63] Klaas-Jan Stol and Brian Fitzgerald. The abc of software engineering research. *ACM Trans. Softw. Eng. Methodol.*, 27(3), September 2018.

[64] Eric Knauss. Constructive master's thesis work in industry: Guidelines for applying design science research, 2021.

[65] John Venable, Jan Pries-Heje, and Richard Baskerville. A comprehensive framework for evaluation in design science research. In Ken Peffers, Marcus Rothenberger, and Bill Kuechler, editors, *Design Science Research in Information Systems. Advances in Theory and Practice*, pages 423–438, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[66] Uwe Flick. *An Introduction to Qualitative Research*. SAGE Publications, London, England, 4 edition, February 2009.

[67] Victoria Clarke and Virginia Braun. *Successful qualitative research*. SAGE Publications, London, England, March 2013.

[68] Bruce L Berg. *Qualitative research methods for the social sciences*. Pearson,

Upper Saddle River, NJ, 7 edition, December 2008.

[69] Jill Jesson, Lydia Matheson, and Fiona M Lacey. *Doing your literature review*. SAGE Publications, London, England, February 2011.

[70] Douglas C Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, Nashville, TN, 8 edition, May 2012.

[71] Matthias Feurer and Frank Hutter. *Hyperparameter Optimization*, pages 3–33. Springer International Publishing, Cham, 2019.

[72] Uwe Flick. *An Introduction to Qualitative Research*. SAGE, New Delhi, India, December 2018.

[73] Rosaline S Barbour. *Doing Focus Groups*. Qualitative Research Kit. SAGE Publications, London, England, 2 edition, October 2018.

[74] Aida Alvinius, Anders Borglund, and Gerry Larsson. *Tematisk analys - Din handbok till fascinerande vetenskap*. 2023.

[75] Ken Peffers, Marcus Rothenberger, Tuure Tuunanen, and Reza Vaezi. Design science research evaluation. In Ken Peffers, Marcus Rothenberger, and Bill Kuechler, editors, *Design Science Research in Information Systems. Advances in Theory and Practice*, pages 398–410, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[76] Jan Pries-Heje, Richard Baskerville, and John Venable. Strategies for design science research evaluation. pages 255–266, 01 2008.

[77] Nancy Carter, Denise Bryant-Lukosius, Alba DiCenso, Jennifer Blythe, and Alan J Neville. The use of triangulation in qualitative research. *Oncol. Nurs. Forum*, 41(5):545–547, September 2014.

[78] Hannu Turtiainen and Andrei Costin. Vulnberta: On automating cwe weakness assignment and improving the quality of cybersecurity cve vulnerabilities through ml/nlp. In *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 618–625, 2024.

[79] Kimia Ameri, Michael Hempel, Hamid Sharif, Juan Lopez, and Kalyan Perumalla. Design of a novel information system for semi-automated management of cybersecurity in industrial control systems. *ACM Trans. Manage. Inf. Syst.*, 14(1), January 2023.

[80] Pragyan KC, Rambod Ghandiparsi, Rocky Slavin, Sepideh Ghanavati, Travis Breaux, and Mitra Bokaei Hosseini. Toward regulatory compliance: A few-shot learning approach to extract processing activities, 2024.

[81] David Rodriguez, Ian Yang, Jose M. Del Alamo, and Norman Sadeh. Large language models: A new approach for privacy policy analysis at scale, 2024.

[82] Kimya Khakzad Shahandashti, Mithila Sivakumar, Mohammad Mahdi Mohajer, Alvine Boaye Belle, Song Wang, and Timothy Lethbridge. Assessing the impact of gpt-4 turbo in generating defeaters for assurance cases. In *Proceedings of the 2024 IEEE/ACM First International Conference on AI Foundation Models and Software Engineering*, FORGE '24, page 52–56, New York, NY, USA, 2024. Association for Computing Machinery.

[83] Elham Rahmani, Nazim H. Madhavji, and Ibtehal Noorwali. Identifying external cross-references using natural language processing. In *Proceedings of the 30th Annual International Conference on Computer Science and Software Engineering*, CASCON '20, page 143–152, USA, 2020. IBM Corp.

[84] Yukang Wang, Yue Liu, Haozhe Cai, Jia Wang, and Xiaoping Zhou. An automated fire code compliance checking jointly using building information models and natural language processing. *Fire*, 6(9):358, September 2023.

[85] Dongming Guo, Erling Onstein, and Angela Daniela La Rosa. A semantic approach for automated rule compliance checking in construction industry. *IEEE Access*, 9:129648–129660, 2021.

[86] Xin Xu and Hubo Cai. Ontology and rule-based natural language processing approach for interpreting textual regulations on underground utility infrastructure. *Advanced Engineering Informatics*, 48:101288, 2021.

[87] Yilun Zhou, Jianjun She, Yixuan Huang, Lingzhi Li, Lei Zhang, and Jiashu Zhang. A design for safety (dfs) semantic framework development based on natural language processing (nlp) for automated compliance checking using bim: The case of china. *Buildings*, 12(6), 2022.

[88] Jin Wu, Xiaorui Xue, and Jiansong Zhang. Invariant signature, logic reasoning, and semantic natural language processing (NLP)-based automated building code compliance checking (I-SNACC) framework. *J. Inf. Technol. Constr.*, 28:1–18, January 2023.

[89] Junlong Peng and Xiangjun Liu. Automated code compliance checking research based on BIM and knowledge graph. *Sci. Rep.*, 13(1):7065, May 2023.

[90] Sihao Li, Jiali Wang, and Zhao Xu. Automated compliance checking for BIM models based on Chinese-NLP and knowledge graph: an integrative conceptual framework. *Eng. Constr. Archit. Manage.*, March 2024.

[91] Yian Chen and Huixian Jiang. Optimizing automated compliance checking with ontology-enhanced natural language processing: Case in the fire safety domain. *J. Environ. Manage.*, 371(123320):123320, December 2024.

[92] Zhe Zheng, Yu-Cheng Zhou, Xin-Zheng Lu, and Jia-Rui Lin. Knowledge-informed semantic alignment and rule interpretation for automated compliance checking. *Automation in Construction*, 142:104524, 2022.

[93] Orlando Amaral Cejas, Muhammad Ilyas Azeem, Sallam Abualhaija, and Lionel C. Briand. Nlp-based automated compliance checking of data processing agreements against gdpr. *IEEE Transactions on Software Engineering*, 49(9):4282–4303, 2023.

[94] Orlando Amaral Cejas, Sallam Abualhaija, and Lionel C. Briand. Compai: A tool for gdpr completeness checking of privacy policies using artificial intelligence. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, ASE '24, page 2366–2369, New York, NY, USA, 2024. Association for Computing Machinery.

[95] Abdel-Jaouad Aberkane, Seppe Vanden Broucke, and Geert Poels. Investigating organizational factors associated with gdpr noncompliance using privacy policies: A machine learning approach. In *2022 IEEE 4th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*, pages 107–113, 2022.

[96] Rajaa El Hamdani, Majd Mustapha, David Restrepo Amariles, Aurore Troussel, Sébastien Meeùs, and Katsiaryna Krasnashchok. A combined rule-based and machine learning approach for automated gdpr compliance checking. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ICAIL '21, page 40–49, New York, NY, USA, 2021. Association for Computing Machinery.

[97] Shuang Liu, Baiyang Zhao, Renjie Guo, Guozhu Meng, Fan Zhang, and Meishan Zhang. Have you been properly notified? automatic compliance analysis of privacy policy text with gdpr article 13. In *Proceedings of the Web Conference 2021*, WWW '21, page 2154–2164, New York, NY, USA, 2021. Association for Computing Machinery.

[98] Damiano Torre, Sallam Abualhaija, Mehrdad Sabetzadeh, Lionel Briand, Katrien Baetens, Peter Goes, and Sylvie Forastier. An ai-assisted approach for checking the completeness of privacy policies against gdpr. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pages 136–146, 2020.

[99] Saka John, Binyamin Adeniyi Ajayi, and Samaila Musa Marafa. Natural language processing and deep learning based techniques for evaluation of companies' privacy policies. In Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Ana Maria A. C. Rocha, and Chiara Garau, editors, *Computational Science and Its Applications – ICCSA 2022 Workshops*, pages 15–32, Cham, 2022. Springer International Publishing.

[100] Oluwafemi Akanfe, Rohit Valecha, and H. Raghav Rao. Design of an inclusive financial privacy index (inf-pie): A financial privacy and digital financial inclusion perspective. *ACM Trans. Manage. Inf. Syst.*, 12(1), December 2020.

[101] Ryan McConkey and Oluwafemi Olukoya. Runtime and design time com-

pleteness checking of dangerous android app permissions against gdpr. *IEEE Access*, 12:1–22, 2024.

[102] Muhammad Sajidur Rahman, Pirouz Naghavi, Blas Kojusner, Sadia Afroz, Byron Williams, Sara Rampazzi, and Vincent Bindschaedler. Permpress: Machine learning-based pipeline to evaluate permissions in app privacy policies. *IEEE Access*, 10:89248–89269, 2022.

[103] Zhengqi Xu, Yixuan Cao, Rongyu Cao, Guoxiang Li, Xuanqiang Liu, Yan Pang, Yangbin Wang, Jianfei Zhang, Allie Cheung, Matthew Tam, Lukas Petrikas, and Ping Luo. Jura: Towards automatic compliance assessment for annual reports of listed companies. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 4264–4272, New York, NY, USA, 2021. Association for Computing Machinery.

[104] Sallam Abualhaija, Chetan Arora, Amin Sleimi, and Lionel C. Briand. Automated question answering for improved understanding of compliance requirements: A multi-document study. In *2022 IEEE 30th International Requirements Engineering Conference (RE)*, pages 39–50, 2022.

[105] Sallam Abualhaija, Chetan Arora, and Lionel C. Briand. Coreqqa: a compliance requirements understanding using question answering tool. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2022, page 1682–1686, New York, NY, USA, 2022. Association for Computing Machinery.

[106] Shohreh Deldari, Mohammad Goudarzi, Aditya Joshi, Arash Shaghaghi, Simon Finn, Flora D. Salim, and Sanjay Jha. Auditnet: A conversational ai-based security assistant [demo], 2024.

[107] Aashna Sofat and Balwinder Sodhi. Speeding up government procurement workflows with LLMs. In *Lecture Notes in Computer Science*, Lecture notes in computer science, pages 27–33. Springer Nature Switzerland, Cham, 2024.

[108] No Kyu Seong, Jae Hee Lee, Jong Beom Lee, and Poong Hyun Seong. Retrieval methodology for similar npp lco cases based on domain specific nlp. *Nuclear Engineering and Technology*, 55(2):421–431, 2023.

[109] Anantaa Kotal, Anupam Joshi, and Karuna Pande Joshi. The effect of text ambiguity on creating policy knowledge graphs. In *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 1491–1500, 2021.

[110] Muhammad Atif Javed, Faiz Ul Muram, and Samina Kanwal. Ontology-based natural language processing for process compliance management. In *Communications in Computer and Information Science*, Communications in computer and information science, pages 309–327. Springer International Publishing,

Cham, 2022.

[111] Yu-Cheng Zhou, Zhe Zheng, Jia-Rui Lin, and Xin-Zheng Lu. Integrating nlp and context-free grammar for complex rule interpretation towards automated compliance checking. *Computers in Industry*, 142:103746, 2022.

[112] Xiaorui Xue and Jiansong Zhang. Building codes part-of-speech tagging performance improvement by error-driven transformational rules. *J. Comput. Civ. Eng.*, 34(5):04020035, September 2020.

[113] IBM Editorial Team. What is a rest api?, n.d. Accessed: 2025-05-06.

[114] Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study, 2024.

[115] UN Regulation No. 155 - Cyber security and cyber security management system. `https://unece.org/transport/documents/2021/03/standards/un-regulation-no-155-cyber-security-and-cyber-security`, 2021.

[116] Catherine Sai, Karolin Winter, Elsa Fernanda, and Stefanie Rinderle-Ma. Detecting deviations between external and internal regulatory requirements for improved process compliance assessment. In *Lecture Notes in Computer Science*, Lecture notes in computer science, pages 401–416. Springer Nature Switzerland, Cham, 2023.

[117] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Legal-bert: The muppets straight out of law school, 2020.

[118] Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. Making large language models a better foundation for dense retrieval, 2023.

[119] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024.

[120] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators, 2020.

[121] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2021.

[122] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev,

Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing

Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham

Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024.

[123] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K.

Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[124] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[125] Ahmed Salman, Sadie Creese, and Michael Goldsmith. Position paper: Leveraging large language models for cybersecurity compliance. In *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 496–503, 2024.

[126] Abdel-Jaouad Aberkane, Geert Poels, and Seppe Vanden Broucke. Exploring automated gdpr-compliance in requirements engineering: A systematic mapping study. *IEEE Access*, 9:66542–66559, 2021.

[127] Peihua Ma, Shawn Tsai, Yiyang He, Xiaoxue Jia, Dongyang Zhen, Ning Yu, Qin Wang, Jaspreet K.C. Ahuja, and Cheng-I Wei. Large language models in food science: Innovations, applications, and future. *Trends in Food Science & Technology*, 148:104488, 2024.

[128] Mirko Locatelli, Elena Seghezzi, Laura Pellegrini, Lavinia Chiara Tagliabue, and Giuseppe Martino Di Giuda. Exploring natural language processing in construction and integration with building information modeling: A scientometric analysis. *Buildings*, 11(12), 2021.

[129] Nanjiang Chen, Xuhui Lin, Hai Jiang, and Yi An. Automated building information modeling compliance check through a large language model combined with deep learning and ontology. *Buildings*, 14(7), 2024.

[130] Hao Li, Rongzheng Yang, Shuangshuang Xu, Yao Xiao, and Hongyu Zhao. Intelligent checking method for construction schemes via fusion of knowledge graph and large language models. *Buildings*, 14(8), 2024.

[131] Wuqiang Shen, Tao Dai, Lei Cui, and Chuanmao Xu. An nlp-based method for assessment of gdpr compliance in data privacy agreements. In *Proceedings of the 2024 4th International Conference on Artificial Intelligence, Automation and High Performance Computing*, AIAHPC '24, page 466–471, New York,

NY, USA, 2024. Association for Computing Machinery.

[132] Ziqiu Zheng, Xuejun Li, He Wang, Gaofei Wu, and Yuqing Zhang. Research on ssl/tls security differences based on rfc documents. In *2024 International Conference on Computing, Networking and Communications (ICNC)*, pages 147–151, 2024.

[133] Kaifa Zhao, Xian Zhan, Le Yu, Shiyao Zhou, Hao Zhou, Xiapu Luo, Haoyu Wang, and Yepang Liu. Demystifying privacy policy of third-party libraries in mobile apps. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 1583–1595, 2023.

[134] Chuan Yan, Mark Huasong Meng, Fuman Xie, and Guangdong Bai. Investigating documented privacy changes in android os. *Proc. ACM Softw. Eng.*, 1(FSE), July 2024.

[135] Tao Lv, Ruishi Li, Yi Yang, Kai Chen, Xiaojing Liao, XiaoFeng Wang, Peiwei Hu, and Luyi Xing. Rtfm! automatic assumption discovery and verification derivation from library document for api misuse detection. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 1837–1852, New York, NY, USA, 2020. Association for Computing Machinery.

[136] Wanita Sherchan, Sue Ann Chen, Simon Harris, Nebula Alam, Khoi-Nguyen Tran, and Christopher J Butler. Cognitive compliance: Assessing regulatory risk in financial advice documents. *Proc. Conf. AAAI Artif. Intell.*, 34(09):13636–13637, April 2020.

[137] Mohammad Sadeq Abolhasani and Rong Pan. Leveraging llm for automated ontology extraction and knowledge graph generation. *arXiv preprint arXiv:2412.00608*, 2024.

[138] Jaeyeol Song, Jin-Kook Lee, Jungsik Choi, and Inhan Kim. Deep learning-based extraction of predicate-argument structure (pas) in building design rule sentences. *Journal of Computational Design and Engineering*, 7(5):563–576, 05 2020.

[139] Fahad Ul Hassan, Tuyen Le, and Chau Le. Automated approach for digitalizing scope of work requirements to support contract management. *J. Constr. Eng. Manag.*, 149(4), April 2023.

[140] João Alberto Da Silva Amaral and Fernando Buarque De Lima Neto. A model for selecting relevant topics in documents aimed at compliance processes. In *2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–6, 2021.

[141] Chetan Arora, John Grundy, Louise Puli, and Natasha Layton. Towards standards-compliant assistive technology product specifications via llms. In

2024 IEEE 32nd International Requirements Engineering Conference Workshops (REW), page 385–389. IEEE, June 2024.

[142] Sallam Abualhaija, Marcello Ceci, Nicolas Sannier, Domenico Bianculli, Lionel C. Briand, Dirk Zetzsche, and Marco Bodellini. Ai-enabled regulatory change analysis of legal requirements. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 5–17, 2024.

[143] Catherine Sai, Shazia Sadiq, Lei Han, Gianluca Demartini, and Stefanie Rinderle-Ma. Which legal requirements are relevant to a business process? comparing AI-driven methods as expert aid. In *Research Challenges in Information Science*, Lecture notes in business information processing, pages 166–182. Springer Nature Switzerland, Cham, 2024.

A

Interview Material in Cycle 1

A.1 Consent Form

Consent and information about processing of personal data in student thesis

I agree to my personal data in the form of: first and last name, company of employment, role at the company as well as audio recordings and their accompanying transcriptions may be treated by Chalmers University of Technology for the study: "Large Language Models for Automating Regulatory Compliance in Software Engineering", where the purpose of the work is to understand what parts of regulatory and certification compliance can be automated, and what challenges there are currently in regulatory and certification compliance. The personal data will be collected for the purpose of understanding the current challenges in regulatory and certification compliance and what parts of these challenges may be automated.

Information

Your personal data will be handled as follows: The data will only be handled by the two master's thesis students, Maximilian Forsell and Eric Erlandsson Hollgren, the supervisor Jan Bosch and the co-supervisor at Malmö University, Helena Holmström Olsson. The data handling will consist of summarizing and analyzing the retrieved data for the purpose of mapping the current challenges in regulatory and certification compliance and in so doing exploring what parts of the challenges may be automated. The information will be stored for the duration of the project and will be permanently deleted once the thesis has been reported and the grades of Maximilian Forsell and Eric Erlandsson Hollgren have been reported in Ladok. Any information present in the printed thesis will be completely anonymized. All information will only be stored on Chalmers' file server. Your consent is valid until further notice. You have the right to withdraw your consent at any time. You do this through contacting forsellm@chalmers.se, hollgren@chalmers.se or registrator@chalmers.se. If you withdraw your consent, we will cease processing personal data we have collected with the support of your consent. Some information may be saved due to Chalmers obligations under Swedish archive legislation. Chalmers University of Technology, org. No. 556479-5598 is personal data controller. You

can find Chalmers privacy policy at www.chalmers.se. As a participant you have the right to receive information about how your personal data is processed. You have the right to have incorrect information corrected, redundant data deleted, request that processing shall be restricted and data transferred to another actor. You also have the right to submit a complaint to the Swedish Authority for Privacy Protection (Integritetsskyddsmyndigheten). Do you have any questions about Chalmers's processing of personal data contact Chalmers's data protection officer at dataskydd@chalmers.se.

I agree that Chalmers University of Technology processes personal data about me in accordance with the above.

Place:	Signature
Date:	Name clarification

A.2 Participant Information Sheet

Researchers information and purpose of the research

We are Eric Erlandsson Hollgren (MSc student at Software Engineering and Technology, Chalmers) and Maximilian Forsell (MSc student at Data Science and AI, Chalmers). We are collecting data for a master's thesis that we are writing on Large Language Models for Automating Regulatory Compliance in Software Engineering (the thesis will be completed in 2025). The data we collect will be used for identification of challenges in the compliance process in industry. More specifically the data will be used as grounds upon which to build a software artifact that can aid in the compliance process. An anonymous redacted version of the data will also be published in the thesis without revealing any part of the internal compliance process and interpretations of a regulation or certificate. This data will merely be used as a way for the researchers to motivate design decisions in the software artifact.

What type of data are being collected

We are collecting data using an expert-interview. An expert-interview is a part of what is called a semi-structured interview meaning that the interview will have prepared open questions and potential follow-up questions. We are interested in hearing about your experience working with compliance of certifications and regulations, what challenges you face and what ideas you have that can be used to solve the challenges. There are no right or wrong answers to any of the questions we ask and what we will discuss!

What participation in the interview will involve

The interview will consist of the expert (interviewee) along with an interviewer and a note taker. It will last about one hour. During the interview you will be asked about challenges, experiences and ideas regarding the compliance process that is employed

in your field of industry. This is to try to answer "RQ1: What are the challenges that software-intensive companies experience in complying with regulations?" as well as potentially provide some insights into "RQ2: What are the opportunities to address these challenges by automating parts of the regulatory compliance process using RAG-systems and LLMs?".

The benefits of taking part

You will get the opportunity to take part in a master's thesis research project and the process. You will aid in the understanding of compliance and the challenges that are faced. The master's thesis will also produce a software artifact that you can use however you want in your compliance process.

Are there any risks involved?

There are no risks or deception involved in this project. To make sure that no data is leaked unintentionally, you will get the chance to read and correct any mistakes in the transcription and the use of the transcription of the interview before the thesis is submitted.

Will I be identifiable?

During the analysis of the interview answers you will be identifiable. The answers will only be handled by the master's thesis students, Maximilian Forsell and Eric Erlandsson Hollgren, the supervisor Jan Bosch, and at Malmö University, the co-supervisor Helena Holmström Olsson. No information gathered will be shared outside of these people. You will not be identifiable in the finished master's thesis. During the analysis part done by Maximilian and Eric you will be identifiable but in the printed thesis you will be completely anonymous.

Can I withdraw from the research?

Yes, once you have agreed to be part of the study, you can still withdraw. If you wish to withdraw, please email Maximilian Forsell or Eric Erlandsson Hollgren as soon as possible, before publication of the master's thesis (submission is in June 2025).

If you have any questions, please contact:

Maximilian Forsell, MSc student at the Department of Computer Science and Engineering, Chalmers University of Technology. Email: forsellm@chalmers.se. Eric Erlandsson Hollgren, MSc student at the Department of Computer Science and Engineering, Chalmers University of Technology. Email: hollgren@chalmers.se.

A.3 Interview Guide

Starting questions

1. What is your name?
2. What company do you work for?
3. What is the name of the department you work in?
4. What is your official role at the company?

Main questions

Background information

1. On a given day, roughly how much time do you spend working with compliance issues?
2. What certificates and/or regulations do you currently work with?
    a Do you primarily work with product or process certificates/regulations?
3. Are you able to give a rough estimation on the overall cost of regulatory compliance within the company?
    a What are the implications of this cost on the company?
4. Do you expect the amount of work spent on compliance in the future to increase, decrease or stay at roughly the same level?
5. Describe roughly the workflow when you develop a product in relation to compliance.
    a What is the process that your company uses to handle regulatory compliance?

Challenges with the process

1. What more specifically do you do in your day to day work, when working with the regulatory compliance process?
    a Could you explain (topic introduced by interviewee) in more detail?
2. In the current process are you actively tracking the newly developed artifacts towards the regulation or certification or is it done on demand?
3. Once a product has been developed, does the process for keeping it compliant change compared to when it was under development?
    a How?
    b What are the reasons for this?

4. What parts of your compliance process take the most amount of time?
    a Is (part introduced by interviewee) also the most challenging?
    b If you would do a rough ranking on the challenges you face, what would that ranking be, from most challenging to least?

Challenges with interpretation

1. What is your experience with *interpreting* certifications and/or regulations?
2. Are there cases where a certification or regulation is contradicting itself?
    a How common is this?
    b How do you handle these cases?
3. If you have problems with interpreting a certification or regulation, how do you deal with this?
4. When a product has to be compliant with *several* certificates or regulations how do you make sure it is compliant?
    a If they are ever conflicting, how do you handle it?
5. Does your company operate in different geographical parts of the world?
    a How do you manage the different compliance processes?

Data and tools

1. What artifacts do you work with in the context of regulatory compliance? Is it code, requirements or some other design artifacts?
    a How is the data being used for compliance structured? (PDFs, Excel-files, git-commits, etc.?)
2. Are there any automation tools used in the process currently?
    a Are any of these tools using AI?
        i Which sort of AI is beings used (i.e LLMs, ML etc)
    b Are the tools you currently use tailored to your data?
    c Which ones work better and which ones work worse?
    d Why is (bad tool introduced by interviewee) bad?
    e Why is (good tool introduced by interviewee) good?

3. If a new AI driven automation tool is adopted, what would you like it to do?
    a In which part of the compliance process will it have the most impact?

Ending questions

1. Is there something you would like to talk about that we have not touched upon?
2. Are you willing to participate in the evaluation of the software artifact? (The evaluation will be in the form of a questionnaire, or interviews for those that are interested.)
3. Are you willing to share the data used for compliance with us given that the use case is for us to build a new tool? This data will not be disclosed with anyone that has not signed an NDA.

B

Demonstration Protocol in Cycle 2

Introduction

Today we will show you a prototype of a system with the aim to be a tool used in compliance tracking and verification. The tool automates the breakdown and mapping of regulations and interpretations onto compliance artifacts. In the context of the system, compliance artifacts are results from different procedures that need to be executed when developing a product. Examples of such procedures could be safety analysis, requirements, and other specifications of a product. The system builds a property graph to aid the tracking across these different artifacts in a structured manner and is used to generate reports on how the compliance process is going with regards to a specific regulation or standard.

Use cases

We envision that our solution has two use cases:

1. When engineers submit a compliance artifact, they will receive instant automatic feedback on what parts of the interpretation/legislation they are compliant with and which parts they are missing.

2. An auditor/manager can get an overview and generate a report of the current status of compliance regarding a legislation, with suggestions on what is missing and how it a greater level of compliance can be reached.

Demo and questions[1]

- Show the entire graph the way it is currently built on main.

- Show a legislation node in the graph, the content of it and how the breakdown works.

  - *Is the content given in the node sufficient to interpret this part of the legislation?*

[1]Steps in the demonstration are written in normal text and questions are italicized. Questions in parenthesis were possibly skipped due to lack of time.

  - *Does automatic breakdown make sense in your eyes?*

- Show an interpretation node connected to the above-mentioned legislation node. Explain how the connection is made, what the content is and how the breakdown works.
  - *Would you, on a general level, agree with this connection? Is the interpretation compliant with that part of the legislation?*
  - *Does connecting a part of an interpretation to a part of the legislation make sense?*
- Show a compliance artifact connected to both the above-mentioned nodes. Explain how the connections are made.
  - *Is it sensible to track a compliance artifact (documentation, requirements) via the interpretation all the way to the legislation?*
- Show the query function in real time.
  - *Is this granular enough? If not, what is missing?*
  - *(Is there something you would want us to focus more on based on the missing data?)*
- Show the requirement breakdown function in real time.
  - *Is this granular enough? If not, what is missing?*
  - *(Is there something you would want us to focus more on based on the missing data?)*

Final questions

- Is it clear how the system works?
- Is it clear what part of the compliance process it is trying to automate?
- Based on what we have described of how the system works, do you think it is a conceptually viable way to structure the data of ISO 26262?
- Based on how you currently do compliance tracking and the current compliance process, would this idea of a system be of any use as a tool?
- How hard would it be to adopt this system into the current compliance process?
  - If yes, why, what can change about the system to ease the implementation?
- Have we missed anything in our implementation?

  - Are there features you would like to see?
  - Are there unnecessary parts?
  - Are there any logical flaws in the current implementation?
- Open question: Is there anything anyone would like to add or ask?

C

Demonstration Protocol in Cycle 3

Introduction

First, a refresher on what the system does and the new scope according to the comments. The new system is more focused on the guideline to standard tracing that we spoke of last time. This proved to be the biggest problem for the system and is therefore why we chose to focus on optimizing it as much as possible.

For the new version of the system we have tried to incorporate as much as possible from the feedback we got in the last evaluation. Show image. Explain what parts have changed and why.

The chunking has been replaced with hierarchical parsing, this means that for the standard we are dividing it into clauses that have work products and subsequently requirements for the work products. For the guidelines we parse them into sections and subsections all the way down to what is called heading 4 in word. We also save the tables in a markdown format that the LLM can use. References inside the documents get relations to one another and for references outside the documents we make a temporary entry that gets replaced if the real document is added later.

We have added a new step in the tracing pipeline that is re-ranking using natural language inference which basically takes the gathered sections and re-ranks the results into a new sorted list. This works differently from the embedding similarity search as the re-ranker provides info if the two texts are entailed contradictory or neutral. We rerank based on the results.

Lastly, we ask the LLM if the inputted guidelines should actually be traced to the requirement.

With the new system we get a significant performance boost. The old system did the trace mapping in 2 hours per guideline and standard. The new system does the same work in 10-20 minutes and does not scale with the number of guidelines already in the system. The system is also able to have 2 different granularity modes, the most granular mode has an LLM step when making the traces and that is when it takes 10-20 minutes but it provides good accuracy. Without the LLM we can have the system do the tracing in <5 minutes but have poorer accuracy.

We need to have human in the loop for this system.

Use cases

We envision that the final version of the system has two use cases:

1. When guideline managers add a new guideline for a standard it can "quickly" be mapped onto the requirements from the standard. The author of the guideline can get fast feedback on where the system feels that the guideline covers the requirements.

2. An auditor/manager can get an overview and generate a report of the current status of guideline coverage regarding a standard, with suggestions on what is missing and how a greater level of guideline coverage can be reached. This enables the company to be more mature in their compliance process. Example IEC 62443 maturity level 2.

Demo and questions

- How do you work with the tracing of guidelines to requirements today? What does the process look like from writing to tracing?

- Do you want the system to be more pessimistic or more optimistic? I.e do you want more arrows or less arrow suggestions generated by the system?

- Show what the standard graph looks like, starting with the source node and tracing it down to clauses, work products, requirements and sub requirements.

  - Is this structure an improvement over the last cycle?

  - Does this structure make sense to you?

- Show what the guideline graph looks like, and how it is broken down into headings and subheadings.

  - Is this structure an improvement over the last cycle?

  - Does this structure make sense to you?

- Illustrate one (or several) traces that the system has made between guideline sections and requirements.

  - Are these traces correct in your opinion? Why/why not?

    * Are there too many?

    * Are there too few?

    * Are they incorrect?

  - Does it make sense to you why the system has made these connections?

Final questions

- Is it clear how the system works?
- Is it clear what part of the compliance process it is trying to automate?
- If the firm was to implement such a tool as a part of its compliance process, do you think it would:
  - Be cost-reducing? Why/why not?
  - Be time-reducing? Why/why not?
  - Increase the accuracy over the current way of doing it? Why/why not?
  - Create value for the firm?
- Have we missed anything in our implementation?
  - Are there features that we have missed that you would like to add?
  - Are there features you dislike and would like to remove or change?
- Open question: Is there anything anyone would like to add or ask?

D

Literature Review

The search phrase was input into Web of Science and ACM Digital Library restricted to work published after 2019, i.e. in the range 2020-2025. The search matched based on title, abstract and keywords. All works identified were briefly reviewed based on abstract, and relevant ones were included in the related works section. Table D.1 illustrates the result of the search.

	Number of results
Web of Science	165
ACM Digital Library	29
Total (excluding duplicates)	182
Relevant	55

Table D.1: Results of the literature search

In the proceeding sections the most relevant works are briefly presented with regard to their domain and what part of the compliance process they are trying to solve. Note that works generally feature a plethora of different NLP-methods for solving the specific problems, including e.g. dependency parsing, parts-of-speech tagging, etc. Details regarding these solutions have been omitted for brevity. Four other relevant works were found as references to the literature search, and were also included in the literature review. The purpose of this is to illustrate possible research gaps, as well as present possible solution concepts which have influenced the authors' design of the system. Table D.2 shows the different themes and the number of articles and their references found in each.

D.1 Review Papers

This thesis is not the first to present a literature review on NLP and compliance automation, as a few other reviews were found in the literature search. However, these generally focused on a narrower aspect with regard to the domain and/or problem solved.

Salman et al. [125] for example focuses, a.o., on how LLMs can be used to overcome some of the current challenges in cybersecurity compliance. In particular,

Theme	Number of articles	References
Review Papers	4	[125], [126], [127], [128]
Question-Answering and Paragraph Location	6	[103], [104], [105], [106], [107], [108]
Compliance Checking	27	[84], [85], [86], [87], [88], [89], [90], [91], [92], [129], [130], [31], [93], [131], [94], [95], [96], [97], [98], [99], [100], [101], [102], [132], [133], [134], [135], [136]
Compliance Structuring	11	[109], [18], [110], [19], [20], [137], [111], [112], [138], [139], [140]
Compliance Tracing	5	[21], [78], [79], [83], [141]
Change Management	2	[37], [142]
Compliance Evidence Generation	3	[80], [81], [82]
Deviation Detection	1	[116]
Regulation Identification	1	[143]

Table D.2: Results of the literature review

the article mentions the challenges: “lack of environmental evidence and coverage”, “static nature of compliance approaches”, “lack of continuous compliance monitoring” and “human subjectivity”, which LLMs could overcome since they can analyze unstructured data, continuously and with regard to changes in both regulation and software, as well as mitigate any factor of human subjectivity. In a similar spirit, Aberkane et al. [126] presents a systematic mapping study of the current literature on the intersection of NLP, GDPR and Requirements Engineering (RE). The main conclusion is that while there is much overlap between NLP and RE, there is less overlap between NLP and GDPR as well as RE and GDPR. Only one article was found that overlapped with all three fields.

Ma et al. [127] on the other hand presents a review of various ways in which LLMs can be applied in food science, with special emphasis on regulatory compliance. With

regards to regulatory compliance, the paper suggests that LLMs can automate the analysis and interpretation of regulatory documents, as they excel in parsing and analyzing and can provide insights on how compliance may be achieved. Specifically, examples are shown of studies using NLP in classification of pharmaceutical injury risks with regards to US Food & Drug and Administration documents, language models for construction of structured knowledge graphs for compliance, and NLP to facilitate the comparison and alignment of regulatory documents with process models. Finally, Locatelli et al. [128] presents a scientometric analysis of NLP and Building Information Modeling (BIM), confirming that much of the current work in this area concerns automated compliance checking.

D.2 Question-Answering and Paragraph Location

A case in point is Xu et al. [103], which develops a simple paragraph location system for compliance checking of annual reports, which is able to reduce the time of compliance checking with 80% compared to the manual approach. The paper uses two ML-models, logistic regression and BERT, trained on a manually annotated dataset. Abualhaija et al. [104], [105] propose similar systems, however as opposed to simply finding the correct paragraph, the systems also tries to generate an answer to the user's question by using pre-trained question-answering BERT-models. In Abualhaija et al. [104], the system is evaluated on four different European regulations, including GDPR and works as a traditional RAG-system. When evaluated on labeled datasets, the system finds the right paragraph in 94% of the cases and the right answer with an accuracy of 91%. Abualhaija et al. [105] elaborates on this system by including a RoBERTa-model for question answering.

In the Internet of Things-security domain, Deldari et al. [106] introduces AuditNet, a conversational AI-assistant based on RAG- and LLM-technology which allows security experts to chat with an AI that can directly retrieve related regulations, standards and policies. The assistant is a regular RAG, with the standards documents parsed, chunked and embedded in a vector database. The query system is somewhat unique, as the system uses NLP-methods to extract which policy and standard the query relates to, and what is the subject of the query. Sofat and Sodhi [107] also proposes a RAG-system but instead focused on question-answering related to the compliance of government procurement processes.

Finally, Seong et al. [108] investigates regulatory compliance in the nuclear power plant domain by developing an NLP-based retriever. The compliance conditions are essentially requirements presented in tabular form, and are structured as various thresholds one must comply with, within certain time spans. The proposed solution takes a user query and does a lookup in a vector database (created using tf-idf) the most similar cases to the query, to help in decision making when complying.

D.3 Compliance Checking

Automated compliance checking is, at least according to this literature search, the most prominent field in the literature. However, it is mostly concentrated to the construction and data privacy domains. The second of these fields will receive a greater focus in this review, since it is inherently closer related to software engineering.

D.3.1 Construction

Automated compliance checking in the construction field is a major research area. Much of the work in this field relies on BIM, which is essentially a digital representation of a building, as well as the fact that the regulations are very clear cut (e.g. "a door of type X may not be within Y feet of an exhaust of type Z"). A general pipeline for automated compliance checking is to use some form of traditional NLP on the given regulation and then convert this into logical rules which can be automatically executed against the BIM. This method, or similar ones, is used in several of the identified works [84], [85], [86]. A number of works also use ontologies and knowledge graphs as a middle step between parsing the document and converting it to logical rules [87], [88], [89], [90], [91], [92].

A more modern approach is presented in Chen et al. [129], which uses one-shot learning with LLMs for structured information extraction from regulatory texts, by first finetuning the LLM on a number of annotated examples. After the structured information has been extracted, compliance checking can be done automatically using a rule-based algorithm, mapping labels to ontology classes and then executing queries. In a similar manner Li et al. [130] demonstrates using LLMs for compliance checking of construction schemes. This method relies on first creating a knowledge graph of a construction standard by analyzing the hierarchy of the construction standard and converting this into graph format. The construction scheme can then be parsed and a BERT-model can be employed to construct cypher statements and query the knowledge graph. The LLM is responsible for constructing cypher statements and then combining the construction scheme text statements with checking points in the knowledge graph to infer whether it is compliant or not, achieving an accuracy of up to 72%.

D.3.2 Data Privacy

In the data privacy and cybersecurity domain, much of the related work regards the completeness checking of data processing agreements (DPAs), privacy policies and app permissions against GDPR. This is likely due to the fact that GDPR gives very clear rules regarding what a DPA or privacy policy must include. For example, if a privacy policy states that it collects personal data then it must also include a statement explaining how the user may request the erasure of such data. In fact, it is accurate to say that any DPA or privacy policy that is complete, i.e. contains all necessary information as stated in GDPR, and is also easy to read, is compliant. Thus, completeness checking is nearly equivalent to compliance checking in this

(albeit niche) domain.

Azeem et al. [31] focuses specifically of completeness checking of DPAs for GDPR, and formalizes the problem succinctly. The formalization states that each sentence in the DPA can be classified as covering one or more provisions in GDPR and if all provisions are covered once all sentences have been classified, then the DPA is complete. The article uses two methods to accomplish this: one binary classifier that checks for each provision if a given sentence is compliant, and one multi-class classifier that checks which provision is covered. The first method allows a single sentence to cover many provisions, whereas the latter allows each sentence to cover only one provision. The authors train various BERT-models for this purpose.

Cejas et al. [93] proposes another way of automating compliance checking of DPAs against GDPR. To create this system, the authors' first, in cooperation with legal experts, built "shall" requirements from GDPR and a glossary of legal concepts in the requirements. The system then checks compliance by first preprocessing the DPA using a plethora of traditional NLP-techniques and then in a rule-based manner compute the number of overlapping words with the GDPR requirements to determine compliance. When evaluated on a dataset of 30 actual DPAs, the system had a precision of 89% and recall of 82%. The result can however be improved with manual verification. In a similar fashion, Shen et al. [131] presents a two-step system: First, text classification is carried out on each sentence of the DPA. That is, each sentence is labeled related to GDPR as e.g. "personal data collection", "purpose of personal data", etc. This labeling is done using an LSTM-network and BERT, which have been trained on a manually annotated dataset. Once the sentences in the policy have been labeled, it can determined whether they are compliant with the main requirements of GDPR by checking them against predetermined rules (e.g. "if label A exists in the policy, then label B must also be present").

As for privacy policies, there are a number of works. Cejas et al. [94] proposes a system that, given a privacy policy, is able semantically analyze its content and, with some input from the user, determine any potential incompleteness violations with an accuracy of 96%. The system is based on a traditional ML-approach where a model has been trained to determine which information type a sentence contains (e.g. "right to access", "right to erasure", etc.), and then rule-based checking allows compliance to be determined. Aberkane et al. [95] uses a somewhat simpler approach to determine if privacy policies are compliant with GDPR. The system is trained on a dataset consisting of around 18 000 sentences labeled according to five GDPR requirements. Once the model is trained, a threshold can be computed for determining how many true positive sentences predicted are necessary for the privacy policy as a whole to be considered compliant. Hamdani et al. [96] proposes a roughly equivalent approach, and Liu et al. [97] also illustrates a very similar approach, however with 10 labels and focusing only on Article 13 in GDPR, using manually created rules to determine which labels are required for compliance. Similarly, Torre et al. [98] introduces a conceptual model to help with compliance checking by defining and specifying metadata pertinent to GDPR in privacy policies and tracing these metadata to articles of GDPR. For evaluating the system, the

authors define completeness criteria, e.g. “metadata types X and Y must be present for the privacy policy to be compliant” etc. The solution could detect 45 out of 47 incompleteness issues, with a precision of 85% and recall of 96%.

Privacy policies have also been investigated by John et al. [99] which uses traditional deep learning algorithms, specifically LSTM, GRU and other neural networks to classify companies' privacy policies with regards to five core requirements of data protection regulations. Based on a large labeled dataset, the authors simply trained these different algorithms to classify privacy policies as compliant or not with regard to the five core requirements, which were manually created. The best average accuracy was achieved by the LSTM model which achieved an F1-score of 0.76-0.85% on the different requirements. Akanfe et al. [100] similarly examines the privacy policies of financial apps and analyzes them using text categorization and topic modeling to create privacy policy compliance scores. The compliance score part works by using first using tf-idf and latent Dirichlet allocations to extract 10 latent topics with 30 words each. The topics were then labeled manually into 10 data privacy dimensions. An aggregate compliance score can then be created by computing the number of data privacy dimension words that are present, divided by the total number of words in a privacy policy.

App permissions, especially Android permissions, have also been the subject of many works in GDPR compliance automation. McConkey and Olukoya [101] presents a method to evaluate the completeness of Android permissions against articles of GDPR, and evaluates it using six different NLP-algorithms. The method works by simply computing the cosine similarity between different parts of the permissions and different articles of GDPR and then greater cosine similarity between two parts indicate greater compliance. Based on this, it was determined that BERT generally performed the best. Moreover, Rahman et al. [102] introduces what they refer to as “PermPress”, an automated system for evaluating the completeness of an Android app's permissions. To develop this system, privacy policies were first annotated manually to create a dataset which maps privacy policy provisions to permissions. Then, various ML approaches were used to predict permissions in a number of annotated privacy policies. Finally, an ML model, specifically a decision tree, was given these annotated datasets and trained to determine if they were complete or not.

There are a few works in the data privacy domain that do not focus on DPAs, privacy policies or app permissions. For example, Zheng et al. [132] introduces an NLP-based tool that extracts rules from RFC documents and assesses their compliance with RFC specifications. RFC 20119 stipulates that all rules most use certain keywords i.e. “MUST”, “SHALL” and so on when a requirement is specified. Thus, NLP can be used to retrieve paragraphs containing these keywords and the semantics of these sentences can then be analyzed to create logical rules which can in turn be turned into test cases with assertions to check compliance. Zhao et al. [133] instead introduces a method for using NLP to analyze whether third-party libraries comply with privacy-related regulations. The system first identifies the third-party libraries data usage algorithmically, and then NLP is used to determine what data

usage is declared in the privacy policy. The system then checks whether the data usage identified is clearly stated in the privacy policy. Finally, Yan et al. [134] focuses on determining whether the Android OS complies with the documented changes in its privacy policy using few-shot prompting and LLMs. A first preprocessing step is required, which is that an ontology must first be manually created based on the documented privacy changes. The ontology can however be populated automatically using rule-based patterns in the text, using tags and Regex. The proposed system then uses an LLM (GPT-4) to generate test cases based on the extracted ontologies, using simple prompts.

D.3.3 Other

Besides construction and data privacy, two other fields attempted to automate compliance checking.

Lv et al. [135] proposes a system to analyze a program's compliance with so-called integrations assumptions, i.e. constraints that a program should comply with when making calls to library APIs. In very broad terms, the proposed system works by first extracting the integration assumptions from various documents using a pre-trained GRU neural network, which is fine-tuned on a manually annotated dataset. To then interpret the discovered integration assumptions, traditional NLP-techniques are used. To finally discover whether the integration assumptions are violated or not the integrations assumptions are first broken into smaller components (creating a tree) which could then be manually mapped to verification code snippets. Sherchan et al. [136] instead explores compliance checking of financial advice against government regulations. The proposed solution is a simple NLP and ML-system which gives a traffic light indication of whether a piece of personal financial advice is compliant with a given regulation according to a number of indicators. With regard to each indicator, a different NLP and ML approach is taken to ensure that the given indicator has been satisfied.

D.4 Compliance Structuring

A number of works focus on processing the regulation into format which is easier to interpret. These solutions do not directly automate compliance checking, but speeds up the process as a whole. Generally, popular methods include using knowledge graphs and ontologies.

D.4.1 Knowledge Graphs and Ontologies for Compliance Structuring

A number of works structure the compliance process as an ontology or a knowledge graph, which is confirmed by Kotal et al. [109]. A knowledge graph-based approach applied to Building Information Modeling (BIM) is presented in Kruiper et al. [18]. Loosely, the system uses traditional NLP-methods to annotate words in regulatory

text as “object”, “action”, “functional”, etc. and then a knowledge graph is generated based on these annotations. Information can then be retrieved from this knowledge graph.

With regards to ontologies, one example is Javed et al. [110] using ontology-based NLP for process compliance management, which is applied in the space software engineering field. The standard is first preprocessed for the purpose of extracting basic syntactic features. Based on these features a manually designed ontology is populated. Second, the standard is used to manually populate the ontology, based on three main categories: “process”, “stakeholders” and “work products”. Third, the processed standard and ontology are matched together by extracting information from the processed standard and combining it with the ontology, the processes and requirements which can be used to generate requirements, processes and mappings in an algorithmic manner. Finally, inconsistencies and gaps between the standard and the requirements and mappings can be identified.

An approach for assessing if a privacy policy is compliant with GDPR is investigated by Elluri et al. [19]. The method uses traditional NLP-methods to extract entities from a privacy policy which are used to populate a knowledge graph. After that, an ML-model was applied to determine if a given GDPR class exists within a privacy policy. A BERT-model was finally used to summarize the text regarding the identified class. A similar approach is introduced by Hua et al. [20]. The paper presents a two stage compliance framework that utilities LLMs and NLI-models. Essentially, corporate policies and related regulatory texts are retrieved using word embeddings and an NLI-model are used to determine if the two documents are contradictory or not. The contradicting corporate regulations and regulatory texts are then fed into a LLM to extract a deeper analysis.

Abolhasani and Pan [137] have developed a framework built with LLMs that first extracts an ontology based on inputs from the user which the system also confirms with the user. After the ontology is confirmed by the user, the system requires some legislation or other text and populates the knowledge graph based on the generated ontology and the provided text.

D.4.2 Other Methods in Compliance Structuring

Zhou et al. [111] further emphasizes that interpreting and processing the regulatory text into a computer-readable format is often the most vital and complex stage in automated compliance checking. To solve this, the regulatory text is preprocessed by selecting relevant sentences, text cleaning and splitting. Then, the words in the preprocessed sentences are labeled as objects, propositions and relations of various kinds and parsed into a syntactic tree. Once the tree has been constructed for each sentence, language specific rules and executable rules can be created.

Similarly, Xue and Zhang [112] presents a method to convert building codes into computable representations using traditional NLP-methods. Song et al. [138], propose a similar solution, using deep learning models to extract predicate-argument

structures from building design rules at the sentence level. Also, Hassan et al. [139] proposes a framework which helps with information extraction from scope of work requirements (i.e. contractual obligations) in the construction industry. In essence, the paper proposes chunking the natural language scope of word requirements and then using traditional NLP-methods to create annotated rules that are easier to follow. Da Silva Amaral and De Lima Neto [140] takes a rather different NLP approach, using topic modeling on European laws and similar documents to find topics to help with information extraction with regard to the compliance process.

D.5 Compliance Tracing

Apart from structuring the regulations, a number of works also focus on tracing/mapping various policies and requirements to the regulation. This is often necessary to prove compliance and valuable as a tool in the compliance process.

One important concept in compliance tracing is crosswalking. Agarwal et al. [21] focuses on developing a semi-automated AI-driven approach for creating crosswalks, i.e. mappings between regulations, policies and requirements. The goal is to take a source document and map these to a set of target documents, such that all requirements mentioned in the source control are covered by the text of the target controls, without including any irrelevant controls. The paper takes a supervised ML approach and fine-tunes a BERT-model on pairs of matching and non-matching control pairs which then must be verified manually. The mappings are stored and presented in a graph database. It should be noted that to map the controls, all possible mappings (i.e. all pairs) are evaluated. Recall achieved is somewhere in the range 78-83% depending on how much manual intervention one requires.

In the cybersecurity domain, Turtiainen and Costin [78] focuses specifically on automation of mapping Common Weakness Enumerations (CWEs) to Common Vulnerabilities and Exposures (CVEs) using NLP. The system uses a labeled dataset which maps CWEs to CVEs to fine-tune BERT-models, achieving accuracies in the range 89-98% depending on the model used. Ameri et al. [79] present a methodology and an algorithm that, using NLP, automatically identifies cybersecurity-related claims in industrial control systems documents. In short, the proposed method takes as input a list of cybersecurity requirements (standards) and a list of vendor-supplied features (specifications and claims) and then compares the features to the requirements to find matches, extended features and violations. This is done by first using a fine-tuned BERT model to detect claims in the vendor supplied features in a multi-classification task.

Although not quite tracing, reference resolution is a conceptually related activity. Rahmani et al. [83] present an NLP-based system which automatically extracts cross-references from project contracts in software engineering and summarizes them for stakeholders. The proposed solution is fairly simple and finds cross-references based on textual and grammatical patterns, which can thus be extracted using Regex.

Finally, and in a rather unrelated field, Arora et al. [141] presents a framework for using LLMs for ensuring that the product specifications of assistive technologies are compliant with standards. In essence the system works by inputting both the standard and product specification, then cleaning, chunking and tokenizing the text and identifying products and categorizes using traditional NLP-methods. Then, keywords from the standard and the product specification are matched against each other based on embeddings and the keywords are compared based on their string similarities against certain thresholds. Finally, a RAG-based LLM gets a hierarchical overview of the standard which is retrieved based on the input product specification, and the LLM then classifies the product into different categories.

D.6 Change Management

One interesting and less explored aspect of compliance is that of change management, i.e. dealing with the evolving dynamic nature of both normative requirements and firm policies. Only two works were found trying to automate parts of this challenge.

Tupsamudre et al. [37] proposes a concept, referred to as "live crosswalks", and a system for managing the aforementioned changes using AI, demonstrated in the cybersecurity domain. The paper suggests that mappings can be created that can trace all the way from regulations, via standards, to requirements and finally code implementations. This is done by assuming that each document $\mathcal{X}$ contains a set of technical requirements $\{x_1, ..., x_n\}$ which in turn can be mapped to a document $\mathcal{Y}$ lower in the hierarchy of documents, with its own technical requirements. The paper does not focus on how these mappings are created, instead it focuses on detecting changes in one document and how these changes propagate through the hierarchy of documents. Change detection is done using an alignment algorithm with the Jaccard index to see how much a technical requirement has changed.

Abualhaija et al. [142] focuses on analyzing the changes in regulatory requirements using AI, and the effect this will have on the subsequent software requirements. The proposed system uses GPTs for this task. Changes between pairs of sentences in different versions of the same regulation are first detected and then fed into an LLM which identifies the textual changes. Then, the pairs of sentences and paragraphs are, via a human analyst, fed into the LLM via a customized prompt to analyze the semantic and deontic changes. Evaluated on a labeled dataset, the system is able to identify textual changes with a precision of 87.8% and recall of 93.5%. The system used GPT-4 as an LLM.

D.7 Compliance Evidence Generation

An important part in staying compliant with regulations is generally also to generate evidence of said compliance. A number of works in the literature has attempted to automate this part of the compliance process. This problem in particular has

garnered much attention with regard to using generative AI, in particular GPTs.

In the field of GDPR compliance, KC et al. [80], focuses on record of processing activities compliance. This requirement essentially states that details regarding data processing activities, e.g. what type of data is handled and why, must be recorded. KC et al. [80] propose a way to use LLMs and few-shot learning to generate segments of records of processing activities based on user-authored usage scenarios. The authors' collected usage scenarios, manually broke them down into records of data processing activities, and then used this data as few-shot learning input and evaluation of the system. Similarly, and with regard to privacy policies, Rodriguez et al. [81] proposes a method for using LLMs to extract privacy practices from privacy policies. In essence, the paper uses both few-shot learning and zero-shot learning with an LLM to extract whether or not a privacy policy affirms that it will extract certain types of personal data, making it essentially several binary classification tasks in one. Using a labeled data set, the authors are able to demonstrate that the system achieves an F_1-score around 93%.

Continuing in the field of software engineering, Khakzad Shahandashti et al. [82] focuses on the automation of defeaters for assurance cases, i.e. structured arguments that allow one to determine if a systems non-functional requirements have been correctly implemented. Defeaters are essentially arguments that challenge the assurance cases, which is a necessary step for ensuring the assurance cases are robust. The proposed method uses an LLM to identify and mitigate defeaters in assurance cases using chain-of-thought prompting, with an expert human-in-the-loop to enhance the reliability.

D.8 Deviation Detection

Companies rarely work directly against external regulatory requirements. Instead, they are transformed into internal requirements, such as policies or guidelines. Detecting deviations between external and internal requirements thus becomes an important issue, as explored in Sai et al. [116] with regards to process compliance. The proposed system takes as input both the regulatory document and the company's realization. Both documents are parsed, preprocessed and filtered. Then, mappings are introduced between the constraints in the regulatory documents and the realization, referred to as "constraint coverage", using word embeddings and cosine similarity. These must inspected manually. In a second step, what deviations are found can be characterized using traditional NLP-methods by breaking sentences down into parts, and depending on which part is the most dissimilar the deviation can be determined. The method is evaluated both on GDPR and ISO 27001. The system achieved an overall deviation detection accuracy of 55%.

D.9 Regulation Identification

Finally, one paper focused on the problem of regulation identification. That is, determining which regulatory requirements an organization must comply with. Sai et al. [143] compares different AI-driven solutions for this task. The paper proposes that there are three ways of solving this problem, namely manual, fully automated using a GPT and semi-automated using embeddings and traditional ML-methods. None of the two latter methods outperform manual analysis, however, in general the LLM-method outperforms the semi-automated method with the caveat that it is less transparent.

E

Full Results from Factorial Experiments in Cycle 3

E.1 Factorial Experiments (Embeddings)

E.1.1 Normal

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	0.9	0,880413704	0,050632911	0,974802016	0,186046512	0,07960199
BERT (ISO)	0.85	0,829347123	0,094936709	0,912886969	0,110294118	0,102040816
BERT (ISO)	0.8	0,745959922	0,170886076	0,81137509	0,093425606	0,120805369
BERT (ISO)	0.75	0,634130575	0,278481013	0,674586033	0,088709677	0,134556575
BERT (ISO)	0.7	0,511312217	0,430379747	0,520518359	0,092643052	0,152466368
BERT (ISO)	0.65	0,405300582	0,64556962	0,377969762	0,105590062	0,181494662
BERT (ISO)	0.6	0,319974144	0,734177215	0,272858171	0,103019538	0,180685358
BERT (ISO)	0.55	0,255979315	0,797468354	0,194384449	0,101204819	0,17961511
BERT (ISO)	0.5	0,212669683	0,867088608	0,138228942	0,102698651	0,183646113
BERT (ISO)	0.45	0,187459599	0,911392405	0,105111591	0,103821197	0,186407767
BERT (ISO)	0.4	0,159017453	0,936708861	0,070554356	0,102849201	0,185347527
BERT (ISO)	0.35	0,138978668	0,96835443	0,044636429	0,103378378	0,186813187
BERT (ISO)	0.3	0,12605042	0,974683544	0,029517639	0,10252996	0,185542169
BERT (ISO)	0.25	0,120232708	0,981012658	0,022318215	0,102445473	0,185517654
Instructor-XL	0.9	0,895927602	0,006329114	0,99712023	0,2	0,012269939
Instructor-XL	0.85	0,894634777	0,025316456	0,993520518	0,307692308	0,046783626
Instructor-XL	0.8	0,892049127	0,056962025	0,987041037	0,333333333	0,097297297
Instructor-XL	0.75	0,879120879	0,094936709	0,968322534	0,254237288	0,138248848
Instructor-XL	0.7	0,851325145	0,151898734	0,930885529	0,2	0,172661871
Instructor-XL	0.65	0,813186813	0,272151899	0,874730022	0,198156682	0,229333333
Instructor-XL	0.6	0,769877182	0,373417722	0,814974802	0,186708861	0,248945148
Instructor-XL	0.55	0,698771816	0,518987342	0,719222462	0,173728814	0,26031746
Instructor-XL	0.5	0,623787977	0,670886076	0,618430526	0,166666667	0,267002519
Instructor-XL	0.45	0,548804137	0,803797468	0,519798416	0,159949622	0,266806723
Instructor-XL	0.4	0,462184874	0,82278481	0,421166307	0,139186296	0,238095238
Instructor-XL	0.35	0,379444085	0,905063291	0,319654428	0,131433824	0,22953451
Instructor-XL	0.3	0,28959276	0,936708861	0,215982721	0,119644301	0,21218638
Instructor-XL	0.25	0,226244344	0,96835443	0,141828654	0,113754647	0,203592814
LegalBERT	0.9	0,551389787	0,696202532	0,534917207	0,145502646	0,240700219
LegalBERT	0.85	0,47188106	0,765822785	0,438444924	0,134295228	0,228517469
LegalBERT	0.8	0,425339367	0,82278481	0,38012959	0,131180626	0,226283725
LegalBERT	0.75	0,395604396	0,841772152	0,344852412	0,127516779	0,221482098
LegalBERT	0.7	0,365869425	0,873417722	0,308135349	0,125568699	0,219570406
LegalBERT	0.65	0,327731092	0,911392405	0,261339093	0,123076923	0,21686747
LegalBERT	0.6	0,296056884	0,917721519	0,225341973	0,118755119	0,210297317
LegalBERT	0.55	0,256625727	0,936708861	0,179265659	0,114906832	0,204702628
LegalBERT	0.5	0,215901745	0,949367089	0,132469402	0,110701107	0,19828156
LegalBERT	0.45	0,186813187	0,949367089	0,100071994	0,107142857	0,192554557
LegalBERT	0.4	0,165481577	0,96835443	0,074154068	0,106323836	0,191609267

Continued on next page

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
LegalBERT	0.35	0,153846154	0,96835443	0,061195104	0,105010295	0,189473684
LegalBERT	0.3	0,142857143	0,993670886	0,046076314	0,105937922	0,191463415
LegalBERT	0.25	0,135746606	1	0,037437005	0,105685619	0,191167574
BERT (ISO+R155)	0.9	0,869424693	0,063291139	0,96112311	0,15625	0,09009009
BERT (ISO+R155)	0.85	0,771816419	0,221518987	0,834413247	0,132075472	0,165484634
BERT (ISO+R155)	0.8	0,580478345	0,436708861	0,596832253	0,109697933	0,175349428
BERT (ISO+R155)	0.75	0,378797673	0,664556962	0,346292297	0,103652517	0,179333903
BERT (ISO+R155)	0.7	0,244343891	0,816455696	0,179265659	0,101654846	0,180798879
BERT (ISO+R155)	0.65	0,1822883	0,917721519	0,098632109	0,103793844	0,186495177
BERT (ISO+R155)	0.6	0,146089205	0,962025316	0,053275738	0,103612815	0,187076923
BERT (ISO+R155)	0.55	0,126696833	0,974683544	0,030237581	0,102598268	0,185654008
BERT (ISO+R155)	0.5	0,120232708	0,993670886	0,02087833	0,103493738	0,187462687
BERT (ISO+R155)	0.45	0,111182935	1	0,010079194	0,103065884	0,186871674
BERT (ISO+R155)	0.4	0,10730446	1	0,005759539	0,102664068	0,186210961
BERT (ISO+R155)	0.35	0,106011635	1	0,004319654	0,102530824	0,18599176
BERT (ISO+R155)	0.3	0,104718811	1	0,00287977	0,102397926	0,185773075
BERT (ISO+R155)	0.25	0,103425986	1	0,001439885	0,102265372	0,185554903
BGE-M3	0.9	0,888170653	0,03164557	0,985601152	0,2	0,054644809
BGE-M3	0.85	0,882999354	0,088607595	0,973362131	0,274509804	0,133971292
BGE-M3	0.8	0,862314156	0,189873418	0,938804896	0,260869565	0,21978022
BGE-M3	0.75	0,809308339	0,291139241	0,86825054	0,200873362	0,237726098
BGE-M3	0.7	0,743374273	0,455696203	0,776097912	0,187989556	0,266173752
BGE-M3	0.65	0,642533937	0,626582278	0,644348452	0,166947723	0,263648469
BGE-M3	0.6	0,5468649	0,765822785	0,521958243	0,154140127	0,256627784
BGE-M3	0.55	0,455074337	0,835443038	0,411807055	0,139093783	0,238482385
BGE-M3	0.5	0,363283775	0,898734177	0,30237581	0,127812781	0,223798266
BGE-M3	0.45	0,27343245	0,936708861	0,197984161	0,117274168	0,208450704
BGE-M3	0.4	0,210730446	0,962025316	0,125269978	0,111192392	0,199344262
BGE-M3	0.35	0,170006464	0,987341772	0,077033837	0,108484006	0,195488722
BGE-M3	0.3	0,142857143	1	0,045356371	0,106469003	0,192448234
BGE-M3	0.25	0,124757595	1	0,025197984	0,104497354	0,189221557

Table E.1: Experiments with max-min normalized threshold

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	2.0	0,897866839	0,006329114	0,999280058	0,5	0,0125
BERT (ISO)	1.75	0,893988365	0,012658228	0,994240461	0,2	0,023809524
BERT (ISO)	1.5	0,884938591	0,044303797	0,980561555	0,205882353	0,072916667
BERT (ISO)	1.25	0,846800259	0,088607595	0,933045356	0,130841121	0,105660377
BERT (ISO)	1.0	0,793148028	0,14556962	0,866810655	0,110576923	0,12568306
BERT (ISO)	0.75	0,712346477	0,196202532	0,771058315	0,088825215	0,122287968
BERT (ISO)	0.5	0,619909502	0,284810127	0,658027358	0,086538462	0,132743363
BERT (ISO)	0.25	0,526179703	0,405063291	0,539956803	0,091038407	0,148664344
BERT (ISO)	0.0	0,441499677	0,563291139	0,427645788	0,100678733	0,170825336
BERT (ISO)	-0.25	0,355526826	0,67721519	0,318934485	0,101614435	0,17671346
BERT (ISO)	-0.5	0,296703297	0,740506329	0,246220302	0,100515464	0,177004539
BERT (ISO)	-0.75	0,253393665	0,803797468	0,190784737	0,101518785	0,180269695
BERT (ISO)	-1.0	0,21719457	0,85443038	0,144708423	0,102040816	0,182309251
BERT (ISO)	-1.25	0,199095023	0,905063291	0,118790497	0,104608632	0,187540984
Instructor-XL	2.0	0,892049127	0,056962025	0,987041037	0,333333333	0,097297297
Instructor-XL	1.75	0,881706529	0,094936709	0,971202304	0,272727273	0,14084507
Instructor-XL	1.5	0,863606981	0,132911392	0,946724262	0,221052632	0,166007905
Instructor-XL	1.25	0,832579186	0,221518987	0,902087833	0,204678363	0,212765957
Instructor-XL	1.0	0,794440853	0,310126582	0,849532037	0,189922481	0,235576923
Instructor-XL	0.75	0,744020685	0,424050633	0,780417567	0,180107527	0,252830189
Instructor-XL	0.5	0,684550743	0,556962025	0,699064075	0,173913043	0,265060241
Instructor-XL	0.25	0,618616677	0,696202532	0,609791217	0,168711656	0,271604938
Instructor-XL	0.0	0,553329024	0,803797468	0,524838013	0,1613723	0,268783069
Instructor-XL	-0.25	0,475759535	0,816455696	0,43700504	0,141602634	0,241347053
Instructor-XL	-0.5	0,407239819	0,886075949	0,352771778	0,134744947	0,233918129
Instructor-XL	-0.75	0,332902392	0,924050633	0,265658747	0,125214408	0,220543807
Instructor-XL	-1.0	0,270200388	0,962025316	0,19150468	0,119215686	0,212142359
Instructor-XL	-1.25	0,209437621	0,974683544	0,122390209	0,112163146	0,201175702
LegalBERT	2.0	0,897866839	0	1	nan	nan

Continued on next page

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
LegalBERT	1.75	0,897866839	0	1	nan	nan
LegalBERT	1.5	0,897866839	0	1	nan	nan
LegalBERT	1.25	0,897866839	0	1	nan	nan
LegalBERT	1.0	0,897866839	0	1	nan	nan
LegalBERT	0.75	0,722042663	0,455696203	0,752339813	0,173076923	0,25087108
LegalBERT	0.5	0,550096962	0,696202532	0,533477322	0,145118734	0,240174672
LegalBERT	0.25	0,468648998	0,784810127	0,432685385	0,135964912	0,231775701
LegalBERT	0.0	0,416289593	0,82278481	0,370050396	0,129353234	0,223559759
LegalBERT	-0.25	0,387201034	0,85443038	0,334053276	0,127358491	0,221674877
LegalBERT	-0.5	0,352294764	0,886075949	0,291576674	0,12455516	0,218408736
LegalBERT	-0.75	0,314802844	0,917721519	0,246220302	0,121644295	0,214814815
LegalBERT	-1.0	0,275371687	0,930379747	0,200863931	0,116945107	0,207773852
LegalBERT	-1.25	0,227537169	0,949367089	0,145428366	0,112191473	0,200668896
BERT (ISO+R155)	2.0	0,895281189	0,025316456	0,994240461	0,333333333	0,047058824
BERT (ISO+R155)	1.75	0,889463478	0,044303797	0,985601152	0,259259259	0,075675676
BERT (ISO+R155)	1.5	0,879120879	0,056962025	0,972642189	0,191489362	0,087804878
BERT (ISO+R155)	1.25	0,855850032	0,094936709	0,942404608	0,157894737	0,118577075
BERT (ISO+R155)	1.0	0,819004525	0,164556962	0,893448524	0,149425287	0,156626506
BERT (ISO+R155)	0.75	0,767291532	0,221518987	0,82937365	0,128676471	0,162790698
BERT (ISO+R155)	0.5	0,703296703	0,291139241	0,750179986	0,117048346	0,166969147
BERT (ISO+R155)	0.25	0,624434389	0,379746835	0,652267819	0,110497238	0,171184023
BERT (ISO+R155)	0.0	0,550743374	0,481012658	0,558675306	0,11030479	0,179456907
BERT (ISO+R155)	-0.25	0,484809308	0,563291139	0,475881929	0,108935129	0,182564103
BERT (ISO+R155)	-0.5	0,405946994	0,639240506	0,379409647	0,104880582	0,180196253
BERT (ISO+R155)	-0.75	0,366515837	0,708860759	0,327573794	0,10707457	0,186046512
BERT (ISO+R155)	-1.0	0,335488041	0,772151899	0,285817135	0,10951526	0,191823899
BERT (ISO+R155)	-1.25	0,308985133	0,791139241	0,254139669	0,107665805	0,189537528
BGE-M3	2.0	0,888817065	0,03164557	0,986321094	0,208333333	0,054945055
BGE-M3	1.75	0,885585003	0,069620253	0,978401728	0,268292683	0,110552764
BGE-M3	1.5	0,87136393	0,120253165	0,956803456	0,240506329	0,160337553
BGE-M3	1.25	0,853910795	0,215189873	0,926565875	0,25	0,231292517
BGE-M3	1.0	0,806076277	0,297468354	0,863930886	0,199152542	0,23857868
BGE-M3	0.75	0,753716871	0,424050633	0,791216703	0,18767507	0,260194175
BGE-M3	0.5	0,672268908	0,582278481	0,6825054	0,17260788	0,266280753
BGE-M3	0.25	0,609566904	0,708860759	0,598272138	0,167164179	0,270531401
BGE-M3	0.0	0,521654816	0,778481013	0,492440605	0,148550725	0,249492901
BGE-M3	-0.25	0,45572075	0,835443038	0,412526998	0,139240506	0,238698011
BGE-M3	-0.5	0,383968972	0,886075949	0,326853852	0,130232558	0,227088402
BGE-M3	-0.75	0,30833872	0,911392405	0,239740821	0,12	0,212076583
BGE-M3	-1.0	0,248868778	0,949367089	0,169186465	0,115030675	0,205198358
BGE-M3	-1.25	0,205559147	0,96835443	0,118790497	0,111111111	0,199348534

Table E.2: Experiments with Z-score threshold

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	1	0,855850032	0,082278481	0,943844492	0,142857143	0,104417671
BERT (ISO)	2	0,803490627	0,113924051	0,881929446	0,098901099	0,105882353
BERT (ISO)	3	0,755009696	0,164556962	0,822174226	0,095238095	0,120649652
BERT (ISO)	4	0,706528765	0,215189873	0,762419006	0,093406593	0,130268199
BERT (ISO)	5	0,658047835	0,265822785	0,702663787	0,092307692	0,137030995
BERT (ISO)	6	0,60310278	0,284810127	0,639308855	0,082417582	0,127840909
BERT (ISO)	7	0,553329024	0,329113924	0,578833693	0,081632653	0,13081761
BERT (ISO)	8	0,504848093	0,379746835	0,519078474	0,082417582	0,135440181
BERT (ISO)	9	0,458952812	0,443037975	0,460763139	0,085470085	0,143295803
BERT (ISO)	10	0,428571429	0,582278481	0,411087113	0,101098901	0,172284644
BERT (ISO)	11	0,389140271	0,67721519	0,35637149	0,106893107	0,184641933
BERT (ISO)	12	0,353587589	0,791139241	0,303815695	0,114468864	0,2
BERT (ISO)	13	0,308985133	0,860759494	0,246220302	0,114961961	0,202833706
BERT (ISO)	14	0,257918552	0,898734177	0,185025198	0,111459969	0,198324022
Instructor-XL	1	0,872656755	0,164556962	0,953203744	0,285714286	0,208835341
Instructor-XL	2	0,826761474	0,227848101	0,894888409	0,197802198	0,211764706
Instructor-XL	3	0,783451842	0,303797468	0,838012959	0,175824176	0,222737819
Instructor-XL	4	0,749191984	0,424050633	0,786177106	0,184065934	0,256704981
Instructor-XL	5	0,714932127	0,544303797	0,734341253	0,189010989	0,280587276

Continued on next page

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
Instructor-XL	6	0,680672269	0,664556962	0,6825054	0,192307692	0,298295455
Instructor-XL	7	0,642533937	0,765822785	0,628509719	0,189952904	0,304402516
Instructor-XL	8	0,60051713	0,848101266	0,572354212	0,184065934	0,30248307
Instructor-XL	9	0,553329024	0,905063291	0,513318934	0,174603175	0,292732856
Instructor-XL	10	0,500969619	0,936708861	0,451403888	0,162637363	0,277153558
Instructor-XL	11	0,442146089	0,936708861	0,385889129	0,147852148	0,25539258
Instructor-XL	12	0,385908209	0,949367089	0,321814255	0,137362637	0,24
Instructor-XL	13	0,32967033	0,962025316	0,257739381	0,128486898	0,226696495
Instructor-XL	14	0,272139625	0,96835443	0,192944564	0,120094192	0,213687151
LegalBERT	1	0,855850032	0,082278481	0,943844492	0,142857143	0,104417671
LegalBERT	2	0,812540401	0,158227848	0,886969042	0,137362637	0,147058824
LegalBERT	3	0,76664512	0,221518987	0,828653708	0,128205128	0,162412993
LegalBERT	4	0,741435036	0,386075949	0,781857451	0,167582418	0,233716475
LegalBERT	5	0,707175178	0,506329114	0,730021598	0,175824176	0,261011419
LegalBERT	6	0,667744021	0,601265823	0,675305976	0,173992674	0,269886364
LegalBERT	7	0,620555915	0,658227848	0,616270698	0,163265306	0,26163522
LegalBERT	8	0,572074984	0,708860759	0,556515479	0,153846154	0,25282167
LegalBERT	9	0,523594053	0,759493671	0,496760259	0,146520147	0,245649949
LegalBERT	10	0,473820297	0,803797468	0,436285097	0,13956044	0,237827715
LegalBERT	11	0,422753717	0,841772152	0,375089993	0,132867133	0,229508197
LegalBERT	12	0,374272786	0,892405063	0,315334773	0,129120879	0,2256
LegalBERT	13	0,325791855	0,943037975	0,255579554	0,125950972	0,222222222
LegalBERT	14	0,269553975	0,955696203	0,19150468	0,118524333	0,210893855
BERT (ISO+R155)	1	0,862314156	0,113924051	0,947444204	0,197802198	0,144578313
BERT (ISO+R155)	2	0,81512605	0,170886076	0,888408927	0,148351648	0,158823529
BERT (ISO+R155)	3	0,770523594	0,240506329	0,830813535	0,139194139	0,176334107
BERT (ISO+R155)	4	0,715578539	0,259493671	0,767458603	0,112637363	0,157088123
BERT (ISO+R155)	5	0,668390433	0,316455696	0,708423326	0,10989011	0,163132137
BERT (ISO+R155)	6	0,618616677	0,360759494	0,647948164	0,104395604	0,161931818
BERT (ISO+R155)	7	0,570135747	0,411392405	0,588192945	0,102040816	0,163522013
BERT (ISO+R155)	8	0,521654816	0,462025316	0,528437725	0,100274725	0,164785553
BERT (ISO+R155)	9	0,475759535	0,525316456	0,47012239	0,101343101	0,169907881
BERT (ISO+R155)	10	0,424692954	0,563291139	0,408927286	0,097802198	0,166666667
BERT (ISO+R155)	11	0,382676147	0,64556962	0,352771778	0,101898102	0,176013805
BERT (ISO+R155)	12	0,334195217	0,696202532	0,293016559	0,100732601	0,176
BERT (ISO+R155)	13	0,298642534	0,810126582	0,240460763	0,108199493	0,190902312
BERT (ISO+R155)	14	0,256625727	0,892405063	0,184305256	0,110675039	0,196927374
BGE-M3	1	0,882999354	0,215189873	0,958963283	0,373626374	0,273092369
BGE-M3	2	0,848739496	0,335443038	0,90712743	0,291208791	0,311764706
BGE-M3	3	0,806722689	0,417721519	0,850971922	0,241758242	0,306264501
BGE-M3	4	0,763413058	0,493670886	0,794096472	0,214285714	0,298850575
BGE-M3	5	0,725274725	0,594936709	0,740100792	0,206593407	0,306688418
BGE-M3	6	0,679379444	0,658227848	0,681785457	0,19047619	0,295454545
BGE-M3	7	0,639948287	0,753164557	0,627069834	0,186813187	0,299371069
BGE-M3	8	0,588881707	0,791139241	0,56587473	0,171703297	0,282167043
BGE-M3	9	0,540400776	0,841772152	0,50611951	0,162393162	0,272262027
BGE-M3	10	0,49062702	0,886075949	0,445644348	0,153846154	0,262172285
BGE-M3	11	0,443438914	0,943037975	0,386609071	0,148851149	0,257118205
BGE-M3	12	0,384615385	0,943037975	0,321094312	0,136446886	0,2384
BGE-M3	13	0,32967033	0,962025316	0,257739381	0,128486898	0,226696495
BGE-M3	14	0,27343245	0,974683544	0,193664507	0,120879121	0,215083799

Table E.3: Experiments with k-based cutoff

E.1.2 Cleaned

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	0.9	0,840982547	0,056962025	0,930165587	0,08490566	0,068181818
BERT (ISO)	0.85	0,731738849	0,202531646	0,791936645	0,099688474	0,133611691
BERT (ISO)	0.8	0,58435682	0,405063291	0,60475162	0,104404568	0,166018158
BERT (ISO)	0.75	0,454427925	0,563291139	0,442044636	0,103009259	0,174168297
BERT (ISO)	0.7	0,352294764	0,664556962	0,316774658	0,099620493	0,173267327
BERT (ISO)	0.65	0,277310924	0,772151899	0,221022318	0,101328904	0,179148311

Continued on next page

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	0.6	0,235294118	0,841772152	0,166306695	0,103020914	0,183574879
BERT (ISO)	0.55	0,196509373	0,873417722	0,119510439	0,101396032	0,181698486
BERT (ISO)	0.5	0,159017453	0,886075949	0,076313895	0,098383696	0,177103099
BERT (ISO)	0.45	0,14479638	0,924050633	0,056155508	0,100205903	0,180804954
BERT (ISO)	0.4	0,137039431	0,955696203	0,043916487	0,102096011	0,184483812
BERT (ISO)	0.35	0,125404008	0,974683544	0,028797696	0,102461743	0,185430464
BERT (ISO)	0.3	0,115061409	0,981012658	0,016558675	0,10190664	0,184633711
BERT (ISO)	0.25	0,107950873	0,981012658	0,008639309	0,101174935	0,183431953
Instructor-XL	0.9	0,896574014	0,012658228	0,99712023	0,333333333	0,024390244
Instructor-XL	0.85	0,895927602	0,025316456	0,994960403	0,363636364	0,047337278
Instructor-XL	0.8	0,890756303	0,063291139	0,98488121	0,322580645	0,105820106
Instructor-XL	0.75	0,874595992	0,101265823	0,962562995	0,235294118	0,14159292
Instructor-XL	0.7	0,844214609	0,164556962	0,921526278	0,192592593	0,177474403
Instructor-XL	0.65	0,790562379	0,265822785	0,85025198	0,168	0,205882353
Instructor-XL	0.6	0,725274725	0,373417722	0,765298776	0,153246753	0,217311234
Instructor-XL	0.55	0,672268908	0,556962025	0,685385169	0,167619048	0,257686676
Instructor-XL	0.5	0,594699418	0,689873418	0,58387329	0,158660844	0,257988166
Instructor-XL	0.45	0,499676794	0,803797468	0,465082793	0,145977011	0,247081712
Instructor-XL	0.4	0,401422107	0,873417722	0,347732181	0,132183908	0,229617304
Instructor-XL	0.35	0,326438268	0,924050633	0,258459323	0,12414966	0,218890555
Instructor-XL	0.3	0,251454428	0,96835443	0,169906407	0,117151608	0,209016393
Instructor-XL	0.25	0,197155785	1	0,105831533	0,112857143	0,202824134
LegalBERT	0.9	0,633484163	0,35443038	0,665226782	0,107485605	0,164948454
LegalBERT	0.85	0,556561086	0,411392405	0,573074154	0,098784195	0,159313725
LegalBERT	0.8	0,51195863	0,449367089	0,519078474	0,096075778	0,158305463
LegalBERT	0.75	0,491919845	0,481012658	0,493160547	0,097435897	0,162046908
LegalBERT	0.7	0,460892049	0,5	0,456443485	0,094724221	0,159274194
LegalBERT	0.65	0,426632191	0,518987342	0,41612671	0,091825308	0,156041865
LegalBERT	0.6	0,393665158	0,569620253	0,373650108	0,09375	0,161001789
LegalBERT	0.55	0,351648352	0,620253165	0,321094312	0,09414025	0,163469558
LegalBERT	0.5	0,306399483	0,715189873	0,259899208	0,099035933	0,173979985
LegalBERT	0.45	0,259857789	0,835443038	0,194384449	0,105515588	0,187366927
LegalBERT	0.4	0,209437621	0,930379747	0,127429806	0,10816777	0,19380356
LegalBERT	0.35	0,171945701	0,987341772	0,079193665	0,108710801	0,195856874
LegalBERT	0.3	0,145442793	0,987341772	0,049676026	0,105691057	0,190942472
LegalBERT	0.25	0,12605042	1	0,026637869	0,104635762	0,189448441
BERT (ISO+R155)	0.9	0,778926955	0,215189873	0,843052556	0,134920635	0,165853659
BERT (ISO+R155)	0.85	0,636069813	0,405063291	0,662347012	0,120075047	0,185238784
BERT (ISO+R155)	0.8	0,48804137	0,556962025	0,480201584	0,108641975	0,181818182
BERT (ISO+R155)	0.75	0,34841629	0,658227848	0,313174946	0,098298677	0,171052632
BERT (ISO+R155)	0.7	0,274078862	0,746835443	0,220302376	0,098251457	0,173657101
BERT (ISO+R155)	0.65	0,221073045	0,803797468	0,154787617	0,097617218	0,174091844
BERT (ISO+R155)	0.6	0,195216548	0,835443038	0,122390209	0,097705403	0,174950298
BERT (ISO+R155)	0.55	0,173884939	0,886075949	0,09287257	0,1	0,179717587
BERT (ISO+R155)	0.5	0,153846154	0,905063291	0,068394528	0,099512874	0,179310345
BERT (ISO+R155)	0.45	0,140271493	0,930379747	0,050395968	0,100272851	0,181034483
BERT (ISO+R155)	0.4	0,124111183	0,936708861	0,031677466	0,09912927	0,179285282
BERT (ISO+R155)	0.35	0,118293471	0,943037975	0,024478042	0,099069149	0,179302046
BERT (ISO+R155)	0.3	0,111829347	0,974683544	0,013678906	0,101049869	0,183115339
BERT (ISO+R155)	0.25	0,109243697	0,987341772	0,009359251	0,101827676	0,184615385
BGE-M3	0.9	0,895927602	0,018987342	0,995680346	0,333333333	0,035928144
BGE-M3	0.85	0,89269554	0,03164557	0,990640749	0,277777778	0,056818182
BGE-M3	0.8	0,878474467	0,063291139	0,971202304	0,2	0,096153846
BGE-M3	0.75	0,856496445	0,14556962	0,937365011	0,209090909	0,171641791
BGE-M3	0.7	0,824175824	0,221518987	0,892728582	0,190217391	0,204678363
BGE-M3	0.65	0,762120233	0,335443038	0,810655148	0,167721519	0,223628692
BGE-M3	0.6	0,696186167	0,455696203	0,723542117	0,157894737	0,234527687
BGE-M3	0.55	0,619909502	0,626582278	0,619150468	0,157643312	0,251908397
BGE-M3	0.5	0,539107951	0,778481013	0,51187905	0,153558052	0,256517205
BGE-M3	0.45	0,447963801	0,848101266	0,402447804	0,139004149	0,23885918
BGE-M3	0.4	0,369747899	0,892405063	0,310295176	0,128298453	0,224343675
BGE-M3	0.35	0,294764059	0,917721519	0,223902088	0,118560916	0,209992759
BGE-M3	0.3	0,226244344	0,987341772	0,139668826	0,115470022	0,206759443
BGE-M3	0.25	0,16354234	0,987341772	0,069834413	0,107734807	0,194271482

Table E.4: Experiments with max-min normalized threshold

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	2.0	0,897866839	0	1	nan	nan
BERT (ISO)	1.75	0,897220427	0	0,999280058	0	nan
BERT (ISO)	1.5	0,893341952	0	0,994960403	0	nan
BERT (ISO)	1.25	0,866839043	0,006329114	0,964722822	0,02	0,009615385
BERT (ISO)	1.0	0,814479638	0,094936709	0,896328294	0,094339623	0,094637224
BERT (ISO)	0.75	0,720103426	0,227848101	0,776097912	0,103746398	0,142574257
BERT (ISO)	0.5	0,605042017	0,348101266	0,634269258	0,097690941	0,152565881
BERT (ISO)	0.25	0,502262443	0,5	0,502519798	0,102597403	0,170258621
BERT (ISO)	0.0	0,417582418	0,588607595	0,39812815	0,100107643	0,171113155
BERT (ISO)	-0.25	0,34324499	0,683544304	0,304535637	0,100558659	0,175324675
BERT (ISO)	-0.5	0,28959276	0,753164557	0,236861051	0,100932994	0,178010471
BERT (ISO)	-0.75	0,253393665	0,829113924	0,187904968	0,104050834	0,184897671
BERT (ISO)	-1.0	0,218487395	0,841772152	0,147588193	0,100987092	0,180338983
BERT (ISO)	-1.25	0,190045249	0,873417722	0,112311015	0,100656455	0,180510137
Instructor-XL	2.0	0,892049127	0,050632911	0,987760979	0,32	0,087431694
Instructor-XL	1.75	0,882999354	0,088607595	0,973362131	0,274509804	0,133971292
Instructor-XL	1.5	0,861667744	0,132911392	0,944564435	0,214285714	0,1640625
Instructor-XL	1.25	0,828700711	0,189873418	0,901367891	0,179640719	0,184615385
Instructor-XL	1.0	0,782159017	0,265822785	0,840892729	0,159695817	0,199524941
Instructor-XL	0.75	0,72656755	0,367088608	0,767458603	0,152230971	0,215213358
Instructor-XL	0.5	0,684550743	0,525316456	0,702663787	0,16733871	0,25382263
Instructor-XL	0.25	0,627666451	0,664556962	0,623470122	0,167197452	0,267175573
Instructor-XL	0.0	0,555268261	0,740506329	0,534197264	0,153141361	0,253796095
Instructor-XL	-0.25	0,464770524	0,841772152	0,421886249	0,142094017	0,243144424
Instructor-XL	-0.5	0,389140271	0,879746835	0,333333333	0,130516432	0,227309894
Instructor-XL	-0.75	0,327731092	0,924050633	0,259899208	0,124361158	0,219219219
Instructor-XL	-1.0	0,266321913	0,96835443	0,186465083	0,119251754	0,212352533
Instructor-XL	-1.25	0,215255333	0,993670886	0,126709863	0,11459854	0,205497382
LegalBERT	2.0	0,897866839	0	1	nan	nan
LegalBERT	1.75	0,897866839	0	1	nan	nan
LegalBERT	1.5	0,897866839	0	1	nan	nan
LegalBERT	1.25	0,897866839	0	1	nan	nan
LegalBERT	1.0	0,738202973	0,240506329	0,794816415	0,117647059	0,158004158
LegalBERT	0.75	0,596638655	0,379746835	0,621310295	0,102389078	0,161290323
LegalBERT	0.5	0,524886878	0,430379747	0,535637149	0,095371669	0,156142365
LegalBERT	0.25	0,494505495	0,462025316	0,498200144	0,094805195	0,157327586
LegalBERT	0.0	0,464124111	0,5	0,460043197	0,095295537	0,160081054
LegalBERT	-0.25	0,421460892	0,518987342	0,410367171	0,091009989	0,154863078
LegalBERT	-0.5	0,385261797	0,582278481	0,362850972	0,094165814	0,162114537
LegalBERT	-0.75	0,333548804	0,651898734	0,297336213	0,095458758	0,166531932
LegalBERT	-1.0	0,277957337	0,797468354	0,218862491	0,104046243	0,184075968
LegalBERT	-1.25	0,21978022	0,924050633	0,139668826	0,108873975	0,194796531
BERT (ISO+R155)	2.0	0,897866839	0	1	nan	nan
BERT (ISO+R155)	1.75	0,897866839	0	1	nan	nan
BERT (ISO+R155)	1.5	0,897220427	0	0,999280058	0	nan
BERT (ISO+R155)	1.25	0,878474467	0,018987342	0,976241901	0,083333333	0,030927835
BERT (ISO+R155)	1.0	0,83387201	0,113924051	0,915766739	0,133333333	0,122866894
BERT (ISO+R155)	0.75	0,747252747	0,291139241	0,799136069	0,141538462	0,19047619
BERT (ISO+R155)	0.5	0,627666451	0,411392405	0,652267819	0,118613139	0,184135977
BERT (ISO+R155)	0.25	0,511312217	0,525316456	0,509719222	0,108638743	0,180043384
BERT (ISO+R155)	0.0	0,411118293	0,632911392	0,385889129	0,104931794	0,180018002
BERT (ISO+R155)	-0.25	0,318681319	0,696202532	0,275737941	0,098566308	0,172684458
BERT (ISO+R155)	-0.5	0,270200388	0,765822785	0,213822894	0,099752679	0,176513494
BERT (ISO+R155)	-0.75	0,230769231	0,797468354	0,166306695	0,098130841	0,174757282
BERT (ISO+R155)	-1.0	0,201680672	0,816455696	0,13174946	0,096629213	0,17280643
BERT (ISO+R155)	-1.25	0,182934712	0,85443038	0,106551476	0,098110465	0,17601043
BGE-M3	2.0	0,890756303	0,03164557	0,988480922	0,238095238	0,055865922
BGE-M3	1.75	0,876535229	0,063291139	0,969042477	0,188679245	0,09478673
BGE-M3	1.5	0,85778927	0,139240506	0,939524838	0,20754717	0,166666667
BGE-M3	1.25	0,833225598	0,221518987	0,902807775	0,205882353	0,213414634
BGE-M3	1.0	0,797026503	0,297468354	0,853851692	0,188	0,230392157
BGE-M3	0.75	0,733031674	0,386075949	0,7724982	0,161803714	0,228037383
BGE-M3	0.5	0,670976083	0,474683544	0,693304536	0,149700599	0,227617602
BGE-M3	0.25	0,609566904	0,651898734	0,60475162	0,15797546	0,254320988
BGE-M3	0.0	0,543632838	0,772151899	0,517638589	0,154040404	0,256842105
BGE-M3	-0.25	0,467356173	0,829113924	0,426205904	0,141163793	0,241252302

Continued on next page

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BGE-M3	-0.5	0,404007757	0,873417722	0,350611951	0,132692308	0,230383973
BGE-M3	-0.75	0,336780866	0,911392405	0,271418287	0,124567474	0,219178082
BGE-M3	-1.0	0,272139625	0,949367089	0,195104392	0,11829653	0,210378682
BGE-M3	-1.25	0,218487395	0,987341772	0,131029518	0,114453412	0,205128205

Table E.5: Experiments with Z-score threshold

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	1	0,849385908	0,050632911	0,94024478	0,087912088	0,064257028
BERT (ISO)	2	0,806076277	0,126582278	0,88336933	0,10989011	0,117647059
BERT (ISO)	3	0,758888171	0,183544304	0,824334053	0,106227106	0,134570766
BERT (ISO)	4	0,709114415	0,227848101	0,763858891	0,098901099	0,137931034
BERT (ISO)	5	0,665804783	0,303797468	0,706983441	0,105494505	0,156606852
BERT (ISO)	6	0,619909502	0,367088608	0,648668107	0,106227106	0,164772727
BERT (ISO)	7	0,571428571	0,417721519	0,588912887	0,103610675	0,166037736
BERT (ISO)	8	0,529411765	0,5	0,532757379	0,108516484	0,178329571
BERT (ISO)	9	0,492566257	0,607594937	0,479481641	0,117216117	0,196519959
BERT (ISO)	10	0,45572075	0,715189873	0,426205904	0,124175824	0,211610487
BERT (ISO)	11	0,413703943	0,797468354	0,370050396	0,125874126	0,217428818
BERT (ISO)	12	0,361344538	0,829113924	0,308135349	0,11996337	0,2096
BERT (ISO)	13	0,307692308	0,85443038	0,24550036	0,114116653	0,201342282
BERT (ISO)	14	0,254040078	0,879746835	0,182865371	0,109105181	0,194134078
Instructor-XL	1	0,87394958	0,170886076	0,953923686	0,296703297	0,21686747
Instructor-XL	2	0,828054299	0,234177215	0,895608351	0,203296703	0,217647059
Instructor-XL	3	0,782159017	0,297468354	0,837293017	0,172161172	0,218097448
Instructor-XL	4	0,74789916	0,417721519	0,785457163	0,181318681	0,252873563
Instructor-XL	5	0,718810601	0,563291139	0,73650108	0,195604396	0,290375204
Instructor-XL	6	0,684550743	0,683544304	0,684665227	0,197802198	0,306818182
Instructor-XL	7	0,646412411	0,784810127	0,630669546	0,19466248	0,311949686
Instructor-XL	8	0,60051713	0,848101266	0,572354212	0,184065934	0,30248307
Instructor-XL	9	0,554621849	0,911392405	0,514038877	0,175824176	0,294779939
Instructor-XL	10	0,500969619	0,936708861	0,451403888	0,162637363	0,277153558
Instructor-XL	11	0,447317388	0,962025316	0,388768898	0,151848152	0,262295082
Instructor-XL	12	0,392372334	0,981012658	0,325413967	0,141941392	0,248
Instructor-XL	13	0,333548804	0,981012658	0,259899208	0,131022823	0,231170768
Instructor-XL	14	0,2760181	0,987341772	0,195104392	0,12244898	0,217877095
LegalBERT	1	0,862314156	0,113924051	0,947444204	0,197802198	0,144578313
LegalBERT	2	0,816418875	0,17721519	0,88912887	0,153846154	0,164705882
LegalBERT	3	0,762766645	0,202531646	0,82649388	0,117216117	0,148491879
LegalBERT	4	0,714285714	0,253164557	0,766738661	0,10989011	0,153256705
LegalBERT	5	0,665804783	0,303797468	0,706983441	0,105494505	0,156606852
LegalBERT	6	0,613445378	0,335443038	0,645068395	0,097069597	0,150568182
LegalBERT	7	0,564964447	0,386075949	0,585313175	0,095761381	0,153459119
LegalBERT	8	0,551389787	0,607594937	0,5449964	0,131868132	0,216704289
LegalBERT	9	0,522301228	0,753164557	0,496040317	0,145299145	0,243602866
LegalBERT	10	0,473820297	0,803797468	0,436285097	0,13956044	0,237827715
LegalBERT	11	0,420168067	0,829113924	0,373650108	0,130869131	0,226056946
LegalBERT	12	0,362637363	0,835443038	0,308855292	0,120879121	0,2112
LegalBERT	13	0,307692308	0,85443038	0,24550036	0,114116653	0,201342282
LegalBERT	14	0,260504202	0,911392405	0,186465083	0,113029827	0,201117318
BERT (ISO+R155)	1	0,858435682	0,094936709	0,945284377	0,164835165	0,120481928
BERT (ISO+R155)	2	0,812540401	0,158227848	0,886969042	0,137362637	0,147058824
BERT (ISO+R155)	3	0,776987718	0,272151899	0,834413247	0,157509158	0,199535963
BERT (ISO+R155)	4	0,734970911	0,35443038	0,778257739	0,153846154	0,214559387
BERT (ISO+R155)	5	0,69424693	0,443037975	0,722822174	0,153846154	0,228384992
BERT (ISO+R155)	6	0,650937298	0,518987342	0,665946724	0,15018315	0,232954545
BERT (ISO+R155)	7	0,601163542	0,563291139	0,605471562	0,139717425	0,223899371
BERT (ISO+R155)	8	0,551389787	0,607594937	0,5449964	0,131868132	0,216704289
BERT (ISO+R155)	9	0,497737557	0,632911392	0,482361411	0,122100122	0,204708291
BERT (ISO+R155)	10	0,449256626	0,683544304	0,422606192	0,118681319	0,202247191
BERT (ISO+R155)	11	0,39948287	0,727848101	0,36213103	0,114885115	0,198446937
BERT (ISO+R155)	12	0,349709114	0,772151899	0,301655868	0,111721612	0,1952
BERT (ISO+R155)	13	0,296056884	0,797468354	0,239020878	0,106508876	0,187919463
BERT (ISO+R155)	14	0,239819005	0,810126582	0,174946004	0,100470958	0,17877095

Continued on next page

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BGE-M3	1	0,876535229	0,183544304	0,955363571	0,318681319	0,232931727
BGE-M3	2	0,834518423	0,265822785	0,899208063	0,230769231	0,247058824
BGE-M3	3	0,789915966	0,335443038	0,841612671	0,194139194	0,245939675
BGE-M3	4	0,74531351	0,405063291	0,784017279	0,175824176	0,245210728
BGE-M3	5	0,69424693	0,443037975	0,722822174	0,153846154	0,228384992
BGE-M3	6	0,661279897	0,569620253	0,671706263	0,164835165	0,255681818
BGE-M3	7	0,61667744	0,639240506	0,614110871	0,15855573	0,25408805
BGE-M3	8	0,577246283	0,734177215	0,559395248	0,159340659	0,261851016
BGE-M3	9	0,536522301	0,82278481	0,503959683	0,158730159	0,266120778
BGE-M3	10	0,489334195	0,879746835	0,444924406	0,152747253	0,260299625
BGE-M3	11	0,435681965	0,905063291	0,382289417	0,142857143	0,246764452
BGE-M3	12	0,384615385	0,943037975	0,321094312	0,136446886	0,2384
BGE-M3	13	0,333548804	0,981012658	0,259899208	0,131022823	0,231170768
BGE-M3	14	0,277310924	0,993670886	0,195824334	0,123233909	0,219273743

Table E.6: Experiments with k-based cutoff

E.1.3 Glossary

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	0.9	0,71040724	0,310126582	0,755939525	0,12628866	0,179487179
BERT (ISO)	0.85	0,627020039	0,379746835	0,655147588	0,111317254	0,172166428
BERT (ISO)	0.8	0,567550097	0,474683544	0,578113751	0,113464448	0,183150183
BERT (ISO)	0.75	0,50678733	0,575949367	0,498920086	0,115628971	0,192592593
BERT (ISO)	0.7	0,449903038	0,594936709	0,433405328	0,106696935	0,180943215
BERT (ISO)	0.65	0,397543633	0,689873418	0,364290857	0,109879032	0,189565217
BERT (ISO)	0.6	0,358758888	0,759493671	0,313174946	0,111731844	0,194805195
BERT (ISO)	0.55	0,325791855	0,791139241	0,272858171	0,110132159	0,193348801
BERT (ISO)	0.5	0,297349709	0,829113924	0,236861051	0,109991604	0,194217939
BERT (ISO)	0.45	0,280542986	0,85443038	0,215262779	0,110204082	0,195227766
BERT (ISO)	0.4	0,261797027	0,867088608	0,192944564	0,108903021	0,193502825
BERT (ISO)	0.35	0,243051067	0,879746835	0,17062635	0,107668474	0,191856453
BERT (ISO)	0.3	0,226890756	0,898734177	0,150467963	0,107413011	0,191891892
BERT (ISO)	0.25	0,215901745	0,911392405	0,136789057	0,107222636	0,191872085
Instructor-XL	0.9	0,705235941	0,474683544	0,731461483	0,167410714	0,247524752
Instructor-XL	0.85	0,625727214	0,715189873	0,615550756	0,174652241	0,280745342
Instructor-XL	0.8	0,503555268	0,841772152	0,465082793	0,151826484	0,257253385
Instructor-XL	0.75	0,416289593	0,873417722	0,364290857	0,135161606	0,234096692
Instructor-XL	0.7	0,36199095	0,911392405	0,29949604	0,128916741	0,225882353
Instructor-XL	0.65	0,325145443	0,943037975	0,254859611	0,125844595	0,222056632
Instructor-XL	0.6	0,294117647	0,962025316	0,218142549	0,122778675	0,217765043
Instructor-XL	0.55	0,246283129	0,962025316	0,164866811	0,115853659	0,206802721
Instructor-XL	0.5	0,21719457	0,96835443	0,13174946	0,112582781	0,201713909
Instructor-XL	0.45	0,21719457	0,96835443	0,13174946	0,112582781	0,201713909
Instructor-XL	0.4	0,184227537	0,981012658	0,093592513	0,109618105	0,197201018
Instructor-XL	0.35	0,180995475	0,993670886	0,088552916	0,110330288	0,198608476
Instructor-XL	0.3	0,177117001	0,993670886	0,084233261	0,10986704	0,197857593
Instructor-XL	0.25	0,170652877	0,993670886	0,077033837	0,109103544	0,19661866
LegalBERT	0.9	0,612798966	0,348101266	0,642908567	0,099818512	0,155148096
LegalBERT	0.85	0,580478345	0,392405063	0,60187185	0,100813008	0,160413972
LegalBERT	0.8	0,559793148	0,411392405	0,576673866	0,099540582	0,160295931
LegalBERT	0.75	0,522301228	0,487341772	0,526277898	0,104761905	0,172452408
LegalBERT	0.7	0,477698772	0,518987342	0,47300216	0,100737101	0,16872428
LegalBERT	0.65	0,443438914	0,537974684	0,432685385	0,097365407	0,164888458
LegalBERT	0.6	0,413703943	0,594936709	0,393088553	0,100320171	0,171689498
LegalBERT	0.55	0,379444085	0,620253165	0,352051836	0,098196393	0,169550173
LegalBERT	0.5	0,351001939	0,639240506	0,318214543	0,096374046	0,167495854
LegalBERT	0.45	0,328377505	0,658227848	0,290856731	0,095500459	0,166800321
LegalBERT	0.4	0,320620556	0,664556962	0,28149748	0,095194923	0,166534496
LegalBERT	0.35	0,303167421	0,670886076	0,261339093	0,093639576	0,164341085
LegalBERT	0.3	0,277957337	0,670886076	0,233261339	0,090520922	0,159518435
LegalBERT	0.25	0,263736264	0,683544304	0,215982721	0,090225564	0,159409594
BERT (ISO+R155)	0.9	0,69424693	0,556962025	0,709863211	0,179226069	0,271186441

Continued on next page

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO+R155)	0.85	0,620555915	0,670886076	0,614830814	0,165366615	0,265331665
BERT (ISO+R155)	0.8	0,561732385	0,797468354	0,534917207	0,163212435	0,270967742
BERT (ISO+R155)	0.75	0,493859082	0,82278481	0,456443485	0,146892655	0,24928092
BERT (ISO+R155)	0.7	0,424692954	0,860759494	0,375089993	0,135458167	0,234079174
BERT (ISO+R155)	0.65	0,370394312	0,873417722	0,313174946	0,126373626	0,2208
BERT (ISO+R155)	0.6	0,298642534	0,886075949	0,231821454	0,115990058	0,205128205
BERT (ISO+R155)	0.55	0,250808016	0,892405063	0,177825774	0,109898675	0,195697432
BERT (ISO+R155)	0.5	0,216548158	0,911392405	0,137508999	0,107302534	0,192
BERT (ISO+R155)	0.45	0,202327085	0,936708861	0,118790497	0,10787172	0,193464052
BERT (ISO+R155)	0.4	0,186813187	0,955696203	0,099352052	0,107703281	0,193589744
BERT (ISO+R155)	0.35	0,168713639	0,974683544	0,077033837	0,10724234	0,193224592
BERT (ISO+R155)	0.3	0,161603103	0,981012658	0,068394528	0,106970324	0,192906036
BERT (ISO+R155)	0.25	0,157724628	0,981012658	0,064074874	0,10652921	0,192188469
BGE-M3	0.9	0,731738849	0,588607595	0,748020158	0,20993228	0,309484193
BGE-M3	0.85	0,620555915	0,715189873	0,609791217	0,172519084	0,27798278
BGE-M3	0.8	0,51195863	0,797468354	0,479481641	0,148409894	0,250248262
BGE-M3	0.75	0,422753717	0,867088608	0,372210223	0,135777998	0,23479006
BGE-M3	0.7	0,371040724	0,911392405	0,309575234	0,130553037	0,228390167
BGE-M3	0.65	0,34324499	0,936708861	0,275737941	0,128249567	0,225609756
BGE-M3	0.6	0,314802844	0,962025316	0,241180706	0,126036484	0,2228739
BGE-M3	0.55	0,277957337	0,96835443	0,199424046	0,120948617	0,215038651
BGE-M3	0.5	0,241111829	0,987341772	0,156227502	0,11746988	0,209959623
BGE-M3	0.45	0,215255333	0,993670886	0,126709863	0,11459854	0,205497382
BGE-M3	0.4	0,186813187	1	0,094312455	0,111581921	0,200762389
BGE-M3	0.35	0,175177763	1	0,081353492	0,110181311	0,198492462
BGE-M3	0.3	0,167420814	1	0,072714183	0,109266943	0,197007481
BGE-M3	0.25	0,164835165	1	0,069834413	0,108965517	0,196517413

Table E.7: Experiments with max-min normalized threshold

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	2.0	0,897866839	0	1	nan	nan
BERT (ISO)	1.75	0,898513251	0,006329114	1	1	0,012578616
BERT (ISO)	1.5	0,897866839	0,006329114	0,999280058	0,5	0,0125
BERT (ISO)	1.25	0,894634777	0,056962025	0,989920806	0,391304348	0,099447514
BERT (ISO)	1.0	0,834518423	0,208860759	0,905687545	0,201219512	0,204968944
BERT (ISO)	0.75	0,728506787	0,335443038	0,773218143	0,144021739	0,201520913
BERT (ISO)	0.5	0,601809955	0,46835443	0,616990641	0,122112211	0,193717277
BERT (ISO)	0.25	0,500969619	0,601265823	0,489560835	0,118159204	0,197505198
BERT (ISO)	0.0	0,420168067	0,721518987	0,385889129	0,117890383	0,202666667
BERT (ISO)	-0.25	0,351648352	0,803797468	0,300215983	0,1155596	0,202068417
BERT (ISO)	-0.5	0,302521008	0,848101266	0,240460763	0,112699748	0,198960653
BERT (ISO)	-0.75	0,270200388	0,873417722	0,201583873	0,110665597	0,196441281
BERT (ISO)	-1.0	0,245636716	0,879746835	0,17350612	0,108003108	0,192387543
BERT (ISO)	-1.25	0,228829994	0,892405063	0,153347732	0,107061503	0,191186441
Instructor-XL	2.0	0,895927602	0	0,997840173	0	nan
Instructor-XL	1.75	0,896574014	0,006329114	0,997840173	0,25	0,012345679
Instructor-XL	1.5	0,893988365	0,012658228	0,994240461	0,2	0,023809524
Instructor-XL	1.25	0,893988365	0,037974684	0,991360691	0,333333333	0,068181818
Instructor-XL	1.0	0,881060116	0,113924051	0,968322534	0,290322581	0,163636364
Instructor-XL	0.75	0,79638009	0,303797468	0,852411807	0,18972332	0,233576642
Instructor-XL	0.5	0,676793794	0,550632911	0,691144708	0,168604651	0,258160237
Instructor-XL	0.25	0,569489334	0,886075949	0,533477322	0,177664975	0,295983087
Instructor-XL	0.0	0,41822883	0,930379747	0,359971202	0,141891892	0,246231156
Instructor-XL	-0.25	0,346477052	0,96835443	0,275737941	0,132010354	0,232346241
Instructor-XL	-0.5	0,306399483	0,96835443	0,231101512	0,125307125	0,221899927
Instructor-XL	-0.75	0,271493213	0,993670886	0,189344852	0,122369447	0,217904233
Instructor-XL	-1.0	0,210084034	1	0,120230382	0,114492754	0,205461638
Instructor-XL	-1.25	0,171945701	1	0,07775378	0,109798471	0,197871008
LegalBERT	2.0	0,895927602	0,050632911	0,992080634	0,421052632	0,09039548
LegalBERT	1.75	0,894634777	0,050632911	0,990640749	0,380952381	0,089385475
LegalBERT	1.5	0,889463478	0,056962025	0,984161267	0,290322581	0,095238095
LegalBERT	1.25	0,862314156	0,063291139	0,953203744	0,133333333	0,08583691
LegalBERT	1.0	0,838396897	0,069620253	0,925845932	0,096491228	0,080882353

Continued on next page

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
LegalBERT	0.75	0,78280543	0,101265823	0,860331174	0,076190476	0,086956522
LegalBERT	0.5	0,652230123	0,386075949	0,6825054	0,121513944	0,184848485
LegalBERT	0.25	0,495151907	0,493670886	0,495320374	0,10012837	0,166488794
LegalBERT	0.0	0,443438914	0,544303797	0,431965443	0,098285714	0,166505324
LegalBERT	-0.25	0,403361345	0,582278481	0,383009359	0,096944152	0,166214995
LegalBERT	-0.5	0,301874596	0,67721519	0,259179266	0,094190141	0,165378671
LegalBERT	-0.75	0,228183581	0,772151899	0,166306695	0,0953125	0,169680111
LegalBERT	-1.0	0,174531351	0,860759494	0,096472282	0,097771387	0,175597159
LegalBERT	-1.25	0,14221073	0,867088608	0,05975522	0,094941095	0,171143036
BERT (ISO+R155)	2.0	0,897866839	0	1	nan	nan
BERT (ISO+R155)	1.75	0,898513251	0,006329114	1	1	0,012578616
BERT (ISO+R155)	1.5	0,896574014	0,006329114	0,997840173	0,25	0,012345679
BERT (ISO+R155)	1.25	0,893988365	0,018987342	0,993520518	0,25	0,035294118
BERT (ISO+R155)	1.0	0,85778927	0,069620253	0,947444204	0,130952381	0,090909091
BERT (ISO+R155)	0.75	0,741435036	0,405063291	0,779697624	0,172972973	0,242424242
BERT (ISO+R155)	0.5	0,641887524	0,702531646	0,634989201	0,17961165	0,286082474
BERT (ISO+R155)	0.25	0,548804137	0,797468354	0,520518359	0,159090909	0,265263158
BERT (ISO+R155)	0.0	0,481577246	0,835443038	0,441324694	0,145374449	0,247654784
BERT (ISO+R155)	-0.25	0,398836458	0,886075949	0,343412527	0,133079848	0,231404959
BERT (ISO+R155)	-0.5	0,30575307	0,892405063	0,239020878	0,11769616	0,207964602
BERT (ISO+R155)	-0.75	0,237233355	0,911392405	0,160547156	0,109923664	0,196185286
BERT (ISO+R155)	-1.0	0,207498384	0,930379747	0,125269978	0,107929515	0,193421053
BERT (ISO+R155)	-1.25	0,199095023	0,962025316	0,112311015	0,109747292	0,197018795
BGE-M3	2.0	0,897866839	0	1	nan	nan
BGE-M3	1.75	0,897220427	0	0,999280058	0	nan
BGE-M3	1.5	0,898513251	0,025316456	0,997840173	0,571428571	0,048484848
BGE-M3	1.25	0,892049127	0,044303797	0,988480922	0,304347826	0,077348066
BGE-M3	1.0	0,881706529	0,101265823	0,970482361	0,280701754	0,148837209
BGE-M3	0.75	0,822236587	0,335443038	0,877609791	0,237668161	0,278215223
BGE-M3	0.5	0,67291532	0,683544304	0,671706263	0,191489362	0,299168975
BGE-M3	0.25	0,513251454	0,860759494	0,473722102	0,156862745	0,265365854
BGE-M3	0.0	0,405300582	0,917721519	0,347012239	0,1378327	0,239669421
BGE-M3	-0.25	0,343891403	0,962025316	0,273578114	0,130921619	0,230477635
BGE-M3	-0.5	0,316095669	0,974683544	0,241180706	0,127483444	0,225475842
BGE-M3	-0.75	0,288299935	0,993670886	0,208063355	0,124900557	0,221908127
BGE-M3	-1.0	0,246929541	0,993670886	0,161987041	0,118849357	0,212305612
BGE-M3	-1.25	0,193277311	1	0,101511879	0,112375533	0,202046036

Table E.8: Experiments with Z-score threshold

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	1	0,867485456	0,139240506	0,950323974	0,241758242	0,176706827
BERT (ISO)	2	0,824175824	0,215189873	0,893448524	0,186813187	0,2
BERT (ISO)	3	0,774402069	0,259493671	0,832973362	0,15018315	0,19025522
BERT (ISO)	4	0,727213963	0,316455696	0,773938085	0,137362637	0,191570881
BERT (ISO)	5	0,681318681	0,379746835	0,71562275	0,131868132	0,195758564
BERT (ISO)	6	0,626373626	0,398734177	0,652267819	0,115384615	0,178977273
BERT (ISO)	7	0,574014221	0,430379747	0,590352772	0,106750392	0,171069182
BERT (ISO)	8	0,53070459	0,506329114	0,533477322	0,10989011	0,180586907
BERT (ISO)	9	0,499030381	0,639240506	0,483081353	0,123321123	0,206755374
BERT (ISO)	10	0,454427925	0,708860759	0,425485961	0,123076923	0,209737828
BERT (ISO)	11	0,40206852	0,740506329	0,363570914	0,116883117	0,201898188
BERT (ISO)	12	0,354880414	0,797468354	0,304535637	0,115384615	0,2016
BERT (ISO)	13	0,303813833	0,835443038	0,243340533	0,111580727	0,196868009
BERT (ISO)	14	0,255332902	0,886075949	0,183585313	0,10989011	0,195530726
Instructor-XL	1	0,87394958	0,170886076	0,953923686	0,296703297	0,21686747
Instructor-XL	2	0,828054299	0,234177215	0,895608351	0,203296703	0,217647059
Instructor-XL	3	0,789915966	0,335443038	0,841612671	0,194139194	0,245939675
Instructor-XL	4	0,741435036	0,386075949	0,781857451	0,167582418	0,233716475
Instructor-XL	5	0,696832579	0,455696203	0,724262059	0,158241758	0,234910277
Instructor-XL	6	0,661279897	0,569620253	0,671706263	0,164835165	0,255681818
Instructor-XL	7	0,636069813	0,734177215	0,624910007	0,182103611	0,291823899
Instructor-XL	8	0,59793148	0,835443038	0,570914327	0,181318681	0,297968397
Instructor-XL	9	0,554621849	0,911392405	0,514038877	0,175824176	0,294779939

Continued on next page

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
Instructor-XL	10	0,500969619	0,936708861	0,451403888	0,162637363	0,277153558
Instructor-XL	11	0,444731739	0,949367089	0,387329014	0,14985015	0,258843831
Instructor-XL	12	0,387201034	0,955696203	0,322534197	0,138278388	0,2416
Instructor-XL	13	0,330963154	0,96835443	0,258459323	0,129332206	0,228187919
Instructor-XL	14	0,274725275	0,981012658	0,194384449	0,12166405	0,216480447
LegalBERT	1	0,855850032	0,082278481	0,943844492	0,142857143	0,104417671
LegalBERT	2	0,803490627	0,113924051	0,881929446	0,098901099	0,105882353
LegalBERT	3	0,751131222	0,14556962	0,820014399	0,084249084	0,106728538
LegalBERT	4	0,698771816	0,17721519	0,758099352	0,076923077	0,107279693
LegalBERT	5	0,647705236	0,215189873	0,696904248	0,074725275	0,110929853
LegalBERT	6	0,604395604	0,291139241	0,640028798	0,084249084	0,130681818
LegalBERT	7	0,572721396	0,424050633	0,589632829	0,105180534	0,168553459
LegalBERT	8	0,526826115	0,487341772	0,531317495	0,105769231	0,173814898
LegalBERT	9	0,479638009	0,544303797	0,472282217	0,105006105	0,17604913
LegalBERT	10	0,424692954	0,563291139	0,408927286	0,097802198	0,166666667
LegalBERT	11	0,387847447	0,670886076	0,355651548	0,105894106	0,182916307
LegalBERT	12	0,331609567	0,683544304	0,291576674	0,098901099	0,1728
LegalBERT	13	0,277957337	0,708860759	0,228941685	0,094674556	0,167039523
LegalBERT	14	0,223012282	0,727848101	0,165586753	0,090266876	0,160614525
BERT (ISO+R155)	1	0,862314156	0,113924051	0,947444204	0,197802198	0,144578313
BERT (ISO+R155)	2	0,821590175	0,202531646	0,892008639	0,175824176	0,188235294
BERT (ISO+R155)	3	0,784744667	0,310126582	0,838732901	0,179487179	0,22737819
BERT (ISO+R155)	4	0,74789916	0,417721519	0,785457163	0,181318681	0,252873563
BERT (ISO+R155)	5	0,708468003	0,512658228	0,730741541	0,178021978	0,264274062
BERT (ISO+R155)	6	0,666451196	0,594936709	0,674586033	0,172161172	0,267045455
BERT (ISO+R155)	7	0,629605688	0,702531646	0,621310295	0,174254317	0,279245283
BERT (ISO+R155)	8	0,581124758	0,753164557	0,561555076	0,163461538	0,268623025
BERT (ISO+R155)	9	0,532643827	0,803797468	0,501799856	0,155067155	0,259979529
BERT (ISO+R155)	10	0,485455721	0,860759494	0,442764579	0,149450549	0,254681648
BERT (ISO+R155)	11	0,427925016	0,867088608	0,377969762	0,136863137	0,236410699
BERT (ISO+R155)	12	0,374272786	0,892405063	0,315334773	0,129120879	0,2256
BERT (ISO+R155)	13	0,318034906	0,905063291	0,251259899	0,120879121	0,213273676
BERT (ISO+R155)	14	0,260504202	0,911392405	0,186465083	0,113029827	0,201117318
BGE-M3	1	0,870071105	0,151898734	0,951763859	0,263736264	0,192771084
BGE-M3	2	0,830639948	0,246835443	0,897048236	0,214285714	0,229411765
BGE-M3	3	0,795087266	0,360759494	0,844492441	0,208791209	0,26450116
BGE-M3	4	0,760827408	0,481012658	0,792656587	0,208791209	0,291187739
BGE-M3	5	0,725274725	0,594936709	0,740100792	0,206593407	0,306688418
BGE-M3	6	0,676793794	0,64556962	0,680345572	0,186813187	0,289772727
BGE-M3	7	0,630898513	0,708860759	0,622030238	0,175824176	0,281761006
BGE-M3	8	0,583710407	0,765822785	0,56299496	0,166208791	0,273137698
BGE-M3	9	0,535229476	0,816455696	0,503239741	0,157509158	0,264073695
BGE-M3	10	0,489334195	0,879746835	0,444924406	0,152747253	0,260299625
BGE-M3	11	0,438267615	0,917721519	0,383729302	0,144855145	0,250215703
BGE-M3	12	0,384615385	0,943037975	0,321094312	0,136446886	0,2384
BGE-M3	13	0,330963154	0,96835443	0,258459323	0,129332206	0,228187919
BGE-M3	14	0,278603749	1	0,196544276	0,124018838	0,220670391

Table E.9: Experiments with k-based cutoff

E.1.4 Hypothetical Document Embeddings

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	0.9	0,880413704	0,018987342	0,978401728	0,090909091	0,031413613
BERT (ISO)	0.85	0,832579186	0,050632911	0,921526278	0,068376068	0,058181818
BERT (ISO)	0.8	0,774402069	0,14556962	0,845932325	0,097046414	0,116455696
BERT (ISO)	0.75	0,680025856	0,234177215	0,730741541	0,090024331	0,130052724
BERT (ISO)	0.7	0,569489334	0,35443038	0,593952484	0,090322581	0,143958869
BERT (ISO)	0.65	0,466709761	0,506329114	0,462203024	0,096735187	0,162436548
BERT (ISO)	0.6	0,374272786	0,651898734	0,342692585	0,101377953	0,175468484
BERT (ISO)	0.55	0,299288946	0,759493671	0,246940245	0,102915952	0,181268882
BERT (ISO)	0.5	0,239172592	0,835443038	0,171346292	0,102883866	0,183206107
BERT (ISO)	0.45	0,199741435	0,892405063	0,120950324	0,103524229	0,185526316

Continued on next page

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	0.4	0,158371041	0,924050633	0,071274298	0,101671309	0,183186951
BERT (ISO)	0.35	0,138332256	0,974683544	0,043196544	0,10384356	0,187690433
BERT (ISO)	0.3	0,126696833	0,981012658	0,029517639	0,103127079	0,186634557
BERT (ISO)	0.25	0,120879121	0,993670886	0,021598272	0,103562005	0,187574671
Instructor-XL	0.9	0,897220427	0,018987342	0,99712023	0,428571429	0,036363636
Instructor-XL	0.85	0,891402715	0,044303797	0,987760979	0,291666667	0,076923077
Instructor-XL	0.8	0,877181642	0,107594937	0,964722822	0,257575758	0,151785714
Instructor-XL	0.75	0,85520362	0,208860759	0,928725702	0,25	0,227586207
Instructor-XL	0.7	0,811893988	0,329113924	0,866810655	0,219409283	0,263291139
Instructor-XL	0.65	0,749838397	0,443037975	0,784737221	0,189701897	0,265654649
Instructor-XL	0.6	0,674854557	0,639240506	0,678905688	0,18464351	0,286524823
Instructor-XL	0.55	0,590820944	0,753164557	0,572354212	0,166900421	0,273249139
Instructor-XL	0.5	0,503555268	0,841772152	0,465082793	0,151826484	0,257253385
Instructor-XL	0.45	0,417582418	0,873417722	0,365730742	0,135426889	0,234494477
Instructor-XL	0.4	0,343891403	0,936708861	0,276457883	0,128360798	0,225781846
Instructor-XL	0.35	0,279250162	0,96835443	0,200863931	0,121140143	0,215341309
Instructor-XL	0.3	0,212669683	0,981012658	0,125269978	0,113138686	0,202879581
Instructor-XL	0.25	0,169360052	0,981012658	0,077033837	0,107863605	0,194357367
LegalBERT	0.9	0,595345831	0,632911392	0,591072714	0,149700599	0,242130751
LegalBERT	0.85	0,511312217	0,683544304	0,491720662	0,132678133	0,222222222
LegalBERT	0.8	0,464770524	0,727848101	0,434845212	0,127777778	0,217391304
LegalBERT	0.75	0,429864253	0,753164557	0,393088553	0,123700624	0,2125
LegalBERT	0.7	0,400129282	0,765822785	0,358531317	0,119565217	0,206837607
LegalBERT	0.65	0,372979961	0,772151899	0,327573794	0,115530303	0,200988468
LegalBERT	0.6	0,339366516	0,784810127	0,288696904	0,111510791	0,195275591
LegalBERT	0.55	0,291531997	0,797468354	0,233981281	0,105882353	0,18694362
LegalBERT	0.5	0,258564964	0,82278481	0,194384449	0,104083267	0,184790334
LegalBERT	0.45	0,226244344	0,835443038	0,156947444	0,101304682	0,180698152
LegalBERT	0.4	0,209437621	0,848101266	0,136789057	0,100525131	0,179745137
LegalBERT	0.35	0,195862961	0,873417722	0,118790497	0,101321586	0,181578947
LegalBERT	0.3	0,170652877	0,924050633	0,084953204	0,10303458	0,185396825
LegalBERT	0.25	0,153199741	0,955696203	0,061915047	0,103851444	0,187344913
BERT (ISO+R155)	0.9	0,898513251	0,037974684	0,996400288	0,545454545	0,071005917
BERT (ISO+R155)	0.85	0,893341952	0,101265823	0,983441325	0,41025641	0,162436548
BERT (ISO+R155)	0.8	0,875242405	0,183544304	0,953923686	0,311827957	0,231075697
BERT (ISO+R155)	0.75	0,844861021	0,316455696	0,904967603	0,274725275	0,294117647
BERT (ISO+R155)	0.7	0,793794441	0,455696203	0,83225342	0,236065574	0,311015119
BERT (ISO+R155)	0.65	0,712992889	0,620253165	0,723542117	0,203319502	0,30625
BERT (ISO+R155)	0.6	0,608274079	0,734177215	0,593952484	0,170588235	0,276849642
BERT (ISO+R155)	0.55	0,514544279	0,841772152	0,477321814	0,154831199	0,261553589
BERT (ISO+R155)	0.5	0,42081448	0,898734177	0,366450684	0,138943249	0,240677966
BERT (ISO+R155)	0.45	0,326438268	0,955696203	0,254859611	0,127318718	0,224702381
BERT (ISO+R155)	0.4	0,242404654	0,987341772	0,157667387	0,117647059	0,210242588
BERT (ISO+R155)	0.35	0,192630899	1	0,100791937	0,112295665	0,201916933
BERT (ISO+R155)	0.3	0,159017453	1	0,063354932	0,108293352	0,195423624
BERT (ISO+R155)	0.25	0,134453782	1	0,03599712	0,105544422	0,190936556
BGE-M3	0.9	0,898513251	0,037974684	0,996400288	0,545454545	0,071005917
BGE-M3	0.85	0,893341952	0,101265823	0,983441325	0,41025641	0,162436548
BGE-M3	0.8	0,875242405	0,183544304	0,953923686	0,311827957	0,231075697
BGE-M3	0.75	0,844861021	0,316455696	0,904967603	0,274725275	0,294117647
BGE-M3	0.7	0,793794441	0,455696203	0,83225342	0,236065574	0,311015119
BGE-M3	0.65	0,712992889	0,620253165	0,723542117	0,203319502	0,30625
BGE-M3	0.6	0,608274079	0,734177215	0,593952484	0,170588235	0,276849642
BGE-M3	0.55	0,514544279	0,841772152	0,477321814	0,154831199	0,261553589
BGE-M3	0.5	0,42081448	0,898734177	0,366450684	0,138943249	0,240677966
BGE-M3	0.45	0,326438268	0,955696203	0,254859611	0,127318718	0,224702381
BGE-M3	0.4	0,242404654	0,987341772	0,157667387	0,117647059	0,210242588
BGE-M3	0.35	0,192630899	1	0,100791937	0,112295665	0,201916933
BGE-M3	0.3	0,159017453	1	0,063354932	0,108293352	0,195423624
BGE-M3	0.25	0,134453782	1	0,03599712	0,105544422	0,190936556

Table E.10: Experiments with max-min normalized threshold

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	2.0	0,894634777	0	0,996400288	0	nan
BERT (ISO)	1.75	0,886877828	0,012658228	0,986321094	0,095238095	0,022346369
BERT (ISO)	1.5	0,867485456	0,03164557	0,962562995	0,087719298	0,046511628
BERT (ISO)	1.25	0,822882999	0,063291139	0,909287257	0,073529412	0,068027211
BERT (ISO)	1.0	0,779573368	0,14556962	0,851691865	0,100436681	0,118863049
BERT (ISO)	0.75	0,700711054	0,221518987	0,755219582	0,093333333	0,131332083
BERT (ISO)	0.5	0,620555915	0,297468354	0,657307415	0,089866157	0,138032305
BERT (ISO)	0.25	0,53070459	0,392405063	0,546436285	0,089595376	0,145882353
BERT (ISO)	0.0	0,452488688	0,512658228	0,445644348	0,095182139	0,160555005
BERT (ISO)	-0.25	0,383968972	0,626582278	0,35637149	0,099697885	0,172024327
BERT (ISO)	-0.5	0,323852618	0,734177215	0,277177826	0,103571429	0,181533646
BERT (ISO)	-0.75	0,272139625	0,810126582	0,210943125	0,104575163	0,185238784
BERT (ISO)	-1.0	0,224305107	0,85443038	0,15262779	0,102896341	0,183673469
BERT (ISO)	-1.25	0,19844861	0,892405063	0,119510439	0,103372434	0,185282523
Instructor-XL	2.0	0,89269554	0,03164557	0,990640749	0,277777778	0,056818182
Instructor-XL	1.75	0,886877828	0,069620253	0,979841613	0,282051282	0,111675127
Instructor-XL	1.5	0,870717518	0,158227848	0,951763859	0,27173913	0,2
Instructor-XL	1.25	0,844214609	0,240506329	0,912886969	0,238993711	0,239747634
Instructor-XL	1.0	0,809954751	0,348101266	0,862491001	0,223577236	0,272277228
Instructor-XL	0.75	0,74789916	0,443037975	0,782577394	0,188172043	0,264150943
Instructor-XL	0.5	0,684550743	0,601265823	0,694024478	0,182692308	0,280235988
Instructor-XL	0.25	0,613445378	0,734177215	0,599712023	0,172619048	0,279518072
Instructor-XL	0.0	0,537168714	0,82278481	0,504679626	0,158924205	0,266393443
Instructor-XL	-0.25	0,459599224	0,841772152	0,41612671	0,140889831	0,24137931
Instructor-XL	-0.5	0,393665158	0,917721519	0,334053276	0,135514019	0,236156352
Instructor-XL	-0.75	0,333548804	0,949367089	0,26349892	0,127877238	0,22539444
Instructor-XL	-1.0	0,2708468	0,96835443	0,19150468	0,119905956	0,213389121
Instructor-XL	-1.25	0,212669683	0,981012658	0,125269978	0,113138686	0,202879581
LegalBERT	2.0	0,897866839	0	1	nan	nan
LegalBERT	1.75	0,897866839	0	1	nan	nan
LegalBERT	1.5	0,897866839	0	1	nan	nan
LegalBERT	1.25	0,897866839	0	1	nan	nan
LegalBERT	1.0	0,894634777	0	0,996400288	0	nan
LegalBERT	0.75	0,67550097	0,481012658	0,69762419	0,153225806	0,232415902
LegalBERT	0.5	0,544925663	0,664556962	0,531317495	0,138888889	0,2297593
LegalBERT	0.25	0,470588235	0,727848101	0,441324694	0,129068462	0,219256435
LegalBERT	0.0	0,426632191	0,759493671	0,388768898	0,123839009	0,212954747
LegalBERT	-0.25	0,396250808	0,765822785	0,354211663	0,118860511	0,205782313
LegalBERT	-0.5	0,357466063	0,778481013	0,309575234	0,113678373	0,198387097
LegalBERT	-0.75	0,299288946	0,791139241	0,243340533	0,106292517	0,187406297
LegalBERT	-1.0	0,255332902	0,82278481	0,190784737	0,103668262	0,184135977
LegalBERT	-1.25	0,217840983	0,835443038	0,147588193	0,100303951	0,179104478
BERT (ISO+R155)	2.0	0,897220427	0,069620253	0,991360691	0,47826087	0,121546961
BERT (ISO+R155)	1.75	0,89010989	0,101265823	0,979841613	0,363636364	0,158415842
BERT (ISO+R155)	1.5	0,875888817	0,17721519	0,955363571	0,311111111	0,225806452
BERT (ISO+R155)	1.25	0,853910795	0,272151899	0,920086393	0,279220779	0,275641026
BERT (ISO+R155)	1.0	0,819650937	0,386075949	0,868970482	0,251028807	0,304239401
BERT (ISO+R155)	0.75	0,765998707	0,487341772	0,797696184	0,215083799	0,298449612
BERT (ISO+R155)	0.5	0,704589528	0,658227848	0,709863211	0,205128205	0,312781955
BERT (ISO+R155)	0.25	0,617970265	0,734177215	0,60475162	0,17443609	0,281895504
BERT (ISO+R155)	0.0	0,543632838	0,82278481	0,51187905	0,160891089	0,269151139
BERT (ISO+R155)	-0.25	0,464124111	0,879746835	0,416846652	0,146469968	0,251129178
BERT (ISO+R155)	-0.5	0,391725921	0,911392405	0,332613391	0,134453782	0,234336859
BERT (ISO+R155)	-0.75	0,316095669	0,955696203	0,243340533	0,12562396	0,222058824
BERT (ISO+R155)	-1.0	0,253393665	0,987341772	0,169906407	0,119174943	0,212678937
BERT (ISO+R155)	-1.25	0,207498384	1	0,117350612	0,11416185	0,204928664
BGE-M3	2.0	0,897220427	0,069620253	0,991360691	0,47826087	0,121546961
BGE-M3	1.75	0,89010989	0,101265823	0,979841613	0,363636364	0,158415842
BGE-M3	1.5	0,875888817	0,17721519	0,955363571	0,311111111	0,225806452
BGE-M3	1.25	0,853910795	0,272151899	0,920086393	0,279220779	0,275641026
BGE-M3	1.0	0,819650937	0,386075949	0,868970482	0,251028807	0,304239401
BGE-M3	0.75	0,765998707	0,487341772	0,797696184	0,215083799	0,298449612
BGE-M3	0.5	0,704589528	0,658227848	0,709863211	0,205128205	0,312781955
BGE-M3	0.25	0,617970265	0,734177215	0,60475162	0,17443609	0,281895504
BGE-M3	0.0	0,543632838	0,82278481	0,51187905	0,160891089	0,269151139
BGE-M3	-0.25	0,464124111	0,879746835	0,416846652	0,146469968	0,251129178

Continued on next page

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BGE-M3	-0.5	0,391725921	0,911392405	0,332613391	0,134453782	0,234336859
BGE-M3	-0.75	0,316095669	0,955696203	0,243340533	0,12562396	0,222058824
BGE-M3	-1.0	0,253393665	0,987341772	0,169906407	0,119174943	0,212678937
BGE-M3	-1.25	0,207498384	1	0,117350612	0,11416185	0,204928664

Table E.11: Experiments with Z-score threshold

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BERT (ISO)	1	0,851971558	0,063291139	0,941684665	0,10989011	0,080321285
BERT (ISO)	2	0,803490627	0,113924051	0,881929446	0,098901099	0,105882353
BERT (ISO)	3	0,753716871	0,158227848	0,821454284	0,091575092	0,116009281
BERT (ISO)	4	0,70782159	0,221518987	0,763138949	0,096153846	0,134099617
BERT (ISO)	5	0,652876535	0,240506329	0,699784017	0,083516484	0,123980424
BERT (ISO)	6	0,606981254	0,303797468	0,641468683	0,087912088	0,136363636
BERT (ISO)	7	0,555914674	0,341772152	0,580273578	0,08477237	0,135849057
BERT (ISO)	8	0,512605042	0,417721519	0,523398128	0,090659341	0,148984199
BERT (ISO)	9	0,473173885	0,512658228	0,468682505	0,098901099	0,165813715
BERT (ISO)	10	0,429864253	0,588607595	0,411807055	0,102197802	0,174157303
BERT (ISO)	11	0,393018746	0,696202532	0,358531317	0,10989011	0,189818809
BERT (ISO)	12	0,349709114	0,772151899	0,301655868	0,111721612	0,1952
BERT (ISO)	13	0,311570782	0,873417722	0,247660187	0,116652578	0,205816555
BERT (ISO)	14	0,260504202	0,911392405	0,186465083	0,113029827	0,201117318
Instructor-XL	1	0,876535229	0,183544304	0,955363571	0,318681319	0,232931727
Instructor-XL	2	0,834518423	0,265822785	0,899208063	0,230769231	0,247058824
Instructor-XL	3	0,791208791	0,341772152	0,842332613	0,197802198	0,250580046
Instructor-XL	4	0,763413058	0,493670886	0,794096472	0,214285714	0,298850575
Instructor-XL	5	0,730446025	0,620253165	0,742980562	0,215384615	0,319738989
Instructor-XL	6	0,688429218	0,702531646	0,686825054	0,203296703	0,315340909
Instructor-XL	7	0,645119586	0,778481013	0,629949604	0,193092622	0,309433962
Instructor-XL	8	0,601809955	0,85443038	0,573074154	0,18543956	0,304740406
Instructor-XL	9	0,554621849	0,911392405	0,514038877	0,175824176	0,294779939
Instructor-XL	10	0,499676794	0,930379747	0,450683945	0,161538462	0,275280899
Instructor-XL	11	0,443438914	0,943037975	0,386609071	0,148851149	0,257118205
Instructor-XL	12	0,387201034	0,955696203	0,322534197	0,138278388	0,2416
Instructor-XL	13	0,332255979	0,974683544	0,259179266	0,130177515	0,229679344
Instructor-XL	14	0,2760181	0,987341772	0,195104392	0,12244898	0,217877095
LegalBERT	1	0,854557207	0,075949367	0,94312455	0,131868132	0,096385542
LegalBERT	2	0,807369101	0,132911392	0,884089273	0,115384615	0,123529412
LegalBERT	3	0,758888171	0,183544304	0,824334053	0,106227106	0,134570766
LegalBERT	4	0,722042663	0,291139241	0,771058315	0,126373626	0,176245211
LegalBERT	5	0,683904331	0,392405063	0,717062635	0,136263736	0,20228385
LegalBERT	6	0,645765999	0,493670886	0,663066955	0,142857143	0,221590909
LegalBERT	7	0,599870718	0,556962025	0,60475162	0,138147567	0,221383648
LegalBERT	8	0,553975436	0,620253165	0,546436285	0,134615385	0,221218962
LegalBERT	9	0,515837104	0,721518987	0,492440605	0,139194139	0,233367451
LegalBERT	10	0,463477699	0,753164557	0,430525558	0,130769231	0,222846442
LegalBERT	11	0,413703943	0,797468354	0,370050396	0,125874126	0,217428818
LegalBERT	12	0,379444085	0,917721519	0,318214543	0,132783883	0,232
LegalBERT	13	0,328377505	0,955696203	0,257019438	0,127641589	0,225205071
LegalBERT	14	0,2708468	0,962025316	0,192224622	0,119309262	0,212290503
BERT (ISO+R155)	1	0,885585003	0,227848101	0,960403168	0,395604396	0,289156627
BERT (ISO+R155)	2	0,861667744	0,398734177	0,914326854	0,346153846	0,370588235
BERT (ISO+R155)	3	0,827407886	0,518987342	0,862491001	0,3003663	0,380510441
BERT (ISO+R155)	4	0,78539108	0,601265823	0,806335493	0,260989011	0,363984674
BERT (ISO+R155)	5	0,745959922	0,696202532	0,75161987	0,241758242	0,358890701
BERT (ISO+R155)	6	0,702650291	0,772151899	0,69474442	0,223443223	0,346590909
BERT (ISO+R155)	7	0,650290886	0,803797468	0,632829374	0,199372057	0,319496855
BERT (ISO+R155)	8	0,60310278	0,860759494	0,573794096	0,186813187	0,306997743
BERT (ISO+R155)	9	0,552036199	0,898734177	0,512598992	0,173382173	0,290685773
BERT (ISO+R155)	10	0,499676794	0,930379747	0,450683945	0,161538462	0,275280899
BERT (ISO+R155)	11	0,447317388	0,962025316	0,388768898	0,151848152	0,262295082
BERT (ISO+R155)	12	0,389786684	0,96835443	0,323974082	0,14010989	0,2448
BERT (ISO+R155)	13	0,333548804	0,981012658	0,259899208	0,131022823	0,231170768
BERT (ISO+R155)	14	0,2760181	0,987341772	0,195104392	0,12244898	0,217877095

Continued on next page

Embedding	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
BGE-M3	1	0,884292178	0,221518987	0,959683225	0,384615385	0,281124498
BGE-M3	2	0,861667744	0,398734177	0,914326854	0,346153846	0,370588235
BGE-M3	3	0,826115061	0,512658228	0,861771058	0,296703297	0,37587007
BGE-M3	4	0,78280543	0,588607595	0,804895608	0,255494505	0,356321839
BGE-M3	5	0,753716871	0,734177215	0,755939525	0,254945055	0,378466558
BGE-M3	6	0,706528765	0,791139241	0,696904248	0,228937729	0,355113636
BGE-M3	7	0,655462185	0,829113924	0,635709143	0,205651491	0,329559748
BGE-M3	8	0,60310278	0,860759494	0,573794096	0,186813187	0,306997743
BGE-M3	9	0,553329024	0,905063291	0,513318934	0,174603175	0,292732856
BGE-M3	10	0,500969619	0,936708861	0,451403888	0,162637363	0,277153558
BGE-M3	11	0,447317388	0,962025316	0,388768898	0,151848152	0,262295082
BGE-M3	12	0,389786684	0,96835443	0,323974082	0,14010989	0,2448
BGE-M3	13	0,333548804	0,981012658	0,259899208	0,131022823	0,231170768
BGE-M3	14	0,2760181	0,987341772	0,195104392	0,12244898	0,217877095

Table E.12: Experiments with k-based cutoff

E.2 Factorial Experiments (Reranker)

Reranker	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
qnli-electra-base	0.9	0,683904331	0,360759494	0,720662347	0,128089888	0,189054726
qnli-electra-base	0.85	0,659987072	0,367088608	0,693304536	0,119834711	0,180685358
qnli-electra-base	0.8	0,640594699	0,367088608	0,671706263	0,112840467	0,172619048
qnli-electra-base	0.75	0,627666451	0,379746835	0,655867531	0,111524164	0,172413793
qnli-electra-base	0.7	0,617970265	0,392405063	0,64362851	0,111310592	0,173426573
qnli-electra-base	0.65	0,606981254	0,405063291	0,629949604	0,110726644	0,173913043
qnli-electra-base	0.6	0,601163542	0,424050633	0,621310295	0,112984823	0,178428762
qnli-electra-base	0.55	0,589528119	0,443037975	0,606191505	0,113452188	0,180645161
qnli-electra-base	0.5	0,58177117	0,449367089	0,596832253	0,11251981	0,179974651
qnli-electra-base	0.45	0,574014221	0,46835443	0,586033117	0,114021572	0,183395291
qnli-electra-base	0.4	0,568842922	0,487341772	0,578113751	0,116138763	0,187576127
qnli-electra-base	0.35	0,561085973	0,506329114	0,567314615	0,117474302	0,190703218
qnli-electra-base	0.3	0,552682612	0,512658228	0,557235421	0,11637931	0,18969555
qnli-electra-base	0.25	0,539754363	0,525316456	0,541396688	0,115277778	0,189066059
qnli-electra-base	0.2	0,527472527	0,556962025	0,524118071	0,117489987	0,194046307
qnli-electra-base	0.15	0,513897867	0,575949367	0,506839453	0,117268041	0,194860814
qnli-electra-base	0.1	0,483516484	0,639240506	0,465802736	0,119810202	0,201798202
qnli-electra-base	0.05	0,430510666	0,721518987	0,397408207	0,119873817	0,205590622
qnli-electra-base	0.0	0,102133161	1	0	0,102133161	0,185337243
nli-deberta-base	0.9	0,895281189	0,037974684	0,992800576	0,375	0,068965517
nli-deberta-base	0.85	0,881060116	0,075949367	0,972642189	0,24	0,115384615
nli-deberta-base	0.8	0,864899806	0,17721519	0,94312455	0,261682243	0,211320755
nli-deberta-base	0.75	0,845507434	0,284810127	0,909287257	0,263157895	0,273556231
nli-deberta-base	0.7	0,826761474	0,481012658	0,866090713	0,290076336	0,361904762
nli-deberta-base	0.65	0,779573368	0,626582278	0,796976242	0,25984252	0,367346939
nli-deberta-base	0.6	0,725921138	0,746835443	0,723542117	0,235059761	0,357575758
nli-deberta-base	0.55	0,612152553	0,803797468	0,590352772	0,182471264	0,297423888
nli-deberta-base	0.5	0,502262443	0,886075949	0,458603312	0,156950673	0,266666667
nli-deberta-base	0.45	0,424046542	0,936708861	0,365730742	0,14382896	0,249368155
nli-deberta-base	0.4	0,347123465	0,955696203	0,277897768	0,13084922	0,230182927
nli-deberta-base	0.35	0,272139625	0,962025316	0,193664507	0,119496855	0,212587413
nli-deberta-base	0.3	0,230122818	0,981012658	0,144708423	0,115413254	0,206528981
nli-deberta-base	0.25	0,164835165	0,981012658	0,07199424	0,10734072	0,193508115
nli-deberta-base	0.2	0,122171946	0,987341772	0,023758099	0,103174603	0,186826347
nli-deberta-base	0.15	0,10989011	0,993670886	0,009359251	0,102413568	0,185688941
nli-deberta-base	0.1	0,105365223	0,993670886	0,004319654	0,101948052	0,184923439
nli-deberta-base	0.05	0,102133161	0,993670886	0,000719942	0,101618123	0,184380505
nli-deberta-base	0.0	0,102133161	1	0	0,102133161	0,185337243
bge-reranker-v2-m3	0.9	0,875888817	0,018987342	0,973362131	0,075	0,03030303
bge-reranker-v2-m3	0.85	0,837750485	0,082278481	0,923686105	0,109243697	0,093862816
bge-reranker-v2-m3	0.8	0,759534583	0,196202532	0,823614111	0,112318841	0,142857143
bge-reranker-v2-m3	0.75	0,634776988	0,379746835	0,663786897	0,113851992	0,175182482

Continued on next page

Reranker	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
bge-reranker-v2-m3	0.7	0,521008403	0,582278481	0,514038877	0,119947849	0,198918919
bge-reranker-v2-m3	0.65	0,425339367	0,702531646	0,393808495	0,116474292	0,199819982
bge-reranker-v2-m3	0.6	0,356173239	0,778481013	0,308135349	0,113468635	0,198067633
bge-reranker-v2-m3	0.55	0,314156432	0,791139241	0,259899208	0,108412836	0,190694127
bge-reranker-v2-m3	0.5	0,277957337	0,841772152	0,213822894	0,108571429	0,192335503
bge-reranker-v2-m3	0.45	0,239172592	0,867088608	0,16774658	0,105955143	0,188835286
bge-reranker-v2-m3	0.4	0,193923723	0,879746835	0,115910727	0,101682516	0,182295082
bge-reranker-v2-m3	0.35	0,157724628	0,911392405	0,07199424	0,100488486	0,181018228
bge-reranker-v2-m3	0.3	0,132514544	0,936708861	0,041036717	0,1	0,180708181
bge-reranker-v2-m3	0.25	0,124111183	0,993670886	0,025197984	0,103904699	0,188136609
bge-reranker-v2-m3	0.2	0,11247576	0,993670886	0,012239021	0,102681491	0,186129223
bge-reranker-v2-m3	0.15	0,104718811	0,993670886	0,003599712	0,101881895	0,184814597
bge-reranker-v2-m3	0.1	0,102779573	1	0,000719942	0,102199224	0,185446009
bge-reranker-v2-m3	0.05	0,102779573	1	0,000719942	0,102199224	0,185446009
bge-reranker-v2-m3	0.0	0,102133161	1	0	0,102133161	0,185337243

Table E.13: Experiments with max-min normalized threshold

Reranker	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
qnli-electra-base	2.0	0,897866839	0	1	nan	nan
qnli-electra-base	1.75	0,897866839	0	1	nan	nan
qnli-electra-base	1.5	0,897866839	0	1	nan	nan
qnli-electra-base	1.25	0,711053652	0,316455696	0,755939525	0,128534704	0,182815356
qnli-electra-base	1.0	0,653522948	0,367088608	0,686105112	0,117408907	0,17791411
qnli-electra-base	0.75	0,625080802	0,386075949	0,652267819	0,112132353	0,173789174
qnli-electra-base	0.5	0,60310278	0,411392405	0,624910007	0,110921502	0,174731183
qnli-electra-base	0.25	0,583063995	0,443037975	0,598992081	0,111642743	0,178343949
qnli-electra-base	0.0	0,570135747	0,487341772	0,579553636	0,116490166	0,188034188
qnli-electra-base	-0.25	0,553329024	0,512658228	0,557955364	0,116546763	0,189917937
qnli-electra-base	-0.5	0,527472527	0,556962025	0,524118071	0,117489987	0,194046307
qnli-electra-base	-0.75	0,479638009	0,651898734	0,460043197	0,120750293	0,203758655
qnli-electra-base	-1.0	0,102133161	1	0	0,102133161	0,185337243
qnli-electra-base	-1.25	0,102133161	1	0	0,102133161	0,185337243
qnli-electra-base	-1.5	0,102133161	1	0	0,102133161	0,185337243
qnli-electra-base	-1.75	0,102133161	1	0	0,102133161	0,185337243
qnli-electra-base	-2.0	0,102133161	1	0	0,102133161	0,185337243
nli-deberta-base	2.0	0,888170653	0,056962025	0,982721382	0,272727273	0,094240838
nli-deberta-base	1.75	0,87394958	0,094936709	0,962562995	0,223880597	0,133333333
nli-deberta-base	1.5	0,860374919	0,196202532	0,935925126	0,258333333	0,223021583
nli-deberta-base	1.25	0,846800259	0,297468354	0,909287257	0,271676301	0,283987915
nli-deberta-base	1.0	0,828054299	0,481012658	0,867530598	0,292307692	0,363636364
nli-deberta-base	0.75	0,787330317	0,601265823	0,80849532	0,263157895	0,366088632
nli-deberta-base	0.5	0,746606335	0,734177215	0,748020158	0,248927039	0,371794872
nli-deberta-base	0.25	0,661279897	0,784810127	0,647228222	0,201954397	0,321243523
nli-deberta-base	0.0	0,555268261	0,85443038	0,521238301	0,16875	0,281837161
nli-deberta-base	-0.25	0,468648998	0,898734177	0,419726422	0,14978903	0,256781193
nli-deberta-base	-0.5	0,398836458	0,936708861	0,337652988	0,138576779	0,241435563
nli-deberta-base	-0.75	0,327731092	0,955696203	0,256299496	0,127533784	0,225037258
nli-deberta-base	-1.0	0,269553975	0,962025316	0,190784737	0,119122257	0,211994421
nli-deberta-base	-1.25	0,23335488	0,981012658	0,148308135	0,115844544	0,207219251
nli-deberta-base	-1.5	0,179056238	0,981012658	0,087832973	0,109001406	0,196202532
nli-deberta-base	-1.75	0,133807369	0,987341772	0,036717063	0,104417671	0,188861985
nli-deberta-base	-2.0	0,113768584	0,993670886	0,013678906	0,102815979	0,186350148
bge-reranker-v2-m3	2.0	0,897220427	0	0,999280058	0	nan
bge-reranker-v2-m3	1.75	0,893341952	0	0,994960403	0	nan
bge-reranker-v2-m3	1.5	0,881060116	0,006329114	0,980561555	0,035714286	0,010752688
bge-reranker-v2-m3	1.25	0,853910795	0,056962025	0,944564435	0,104651163	0,073770492
bge-reranker-v2-m3	1.0	0,805429864	0,139240506	0,881209503	0,117647059	0,127536232
bge-reranker-v2-m3	0.75	0,717517776	0,291139241	0,766018719	0,123989218	0,173913043
bge-reranker-v2-m3	0.5	0,613445378	0,430379747	0,634269258	0,118055556	0,185286104
bge-reranker-v2-m3	0.25	0,51195863	0,588607595	0,503239741	0,118773946	0,197662062
bge-reranker-v2-m3	0.0	0,429217841	0,702531646	0,39812815	0,117212249	0,200904977
bge-reranker-v2-m3	-0.25	0,366515837	0,772151899	0,32037437	0,114446529	0,199346405
bge-reranker-v2-m3	-0.5	0,32449903	0,791139241	0,271418287	0,109938434	0,193050193

Continued on next page

Reranker	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
bge-reranker-v2-m3	-0.75	0,292824822	0,803797468	0,234701224	0,106722689	0,1884273
bge-reranker-v2-m3	-1.0	0,261797027	0,85443038	0,194384449	0,107655502	0,19121813
bge-reranker-v2-m3	-1.25	0,222365869	0,867088608	0,149028078	0,103866566	0,185511171
bge-reranker-v2-m3	-1.5	0,186166774	0,879746835	0,107271418	0,100797679	0,180871828
bge-reranker-v2-m3	-1.75	0,156431803	0,911392405	0,070554356	0,100348432	0,18079096
bge-reranker-v2-m3	-2.0	0,133807369	0,936708861	0,042476602	0,100135318	0,180929095

Table E.14: Experiments with Z-score threshold

Reranker	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
qnli-electra-base	1	0,866192631	0,132911392	0,949604032	0,230769231	0,168674699
qnli-electra-base	2	0,821590175	0,202531646	0,892008639	0,175824176	0,188235294
qnli-electra-base	3	0,773109244	0,253164557	0,83225342	0,146520147	0,185614849
qnli-electra-base	4	0,724628313	0,303797468	0,7724982	0,131868132	0,183908046
qnli-electra-base	5	0,676147382	0,35443038	0,712742981	0,123076923	0,182707993
qnli-electra-base	6	0,628959276	0,411392405	0,653707703	0,119047619	0,184659091
qnli-electra-base	7	0,588235294	0,5	0,598272138	0,124018838	0,198742138
qnli-electra-base	8	0,546218487	0,582278481	0,542116631	0,126373626	0,207674944
qnli-electra-base	9	0,50937298	0,689873418	0,488840893	0,133089133	0,223132037
qnli-electra-base	10	0,463477699	0,753164557	0,430525558	0,130769231	0,222846442
qnli-electra-base	11	0,416289593	0,810126582	0,371490281	0,127872128	0,220880069
qnli-electra-base	12	0,363930187	0,841772152	0,309575234	0,121794872	0,2128
qnli-electra-base	13	0,316742081	0,898734177	0,250539957	0,120033812	0,211782252
qnli-electra-base	14	0,266968326	0,943037975	0,190064795	0,116954474	0,208100559
qnli-electra-base	15	0,21460892	0,974683544	0,128149748	0,112820513	0,202232436
qnli-electra-base	16	0,159663866	0,993670886	0,064794816	0,10782967	0,194547708
qnli-electra-base	17	0,102133161	1	0	0,102133161	0,185337243
nli-deberta-base	1	0,87394958	0,170886076	0,953923686	0,296703297	0,21686747
nli-deberta-base	2	0,850032321	0,341772152	0,907847372	0,296703297	0,317647059
nli-deberta-base	3	0,83904331	0,575949367	0,868970482	0,333333333	0,422273782
nli-deberta-base	4	0,806076277	0,702531646	0,817854572	0,304945055	0,425287356
nli-deberta-base	5	0,760180995	0,765822785	0,759539237	0,265934066	0,394779772
nli-deberta-base	6	0,70782159	0,797468354	0,69762419	0,230769231	0,357954545
nli-deberta-base	7	0,65675501	0,835443038	0,636429086	0,20722135	0,332075472
nli-deberta-base	8	0,606981254	0,879746835	0,575953924	0,190934066	0,313769752
nli-deberta-base	9	0,554621849	0,911392405	0,514038877	0,175824176	0,294779939
nli-deberta-base	10	0,502262443	0,943037975	0,45212383	0,163736264	0,279026217
nli-deberta-base	11	0,446024564	0,955696203	0,388048956	0,150849151	0,260569456
nli-deberta-base	12	0,388493859	0,962025316	0,32325414	0,139194139	0,2432
nli-deberta-base	13	0,332255979	0,974683544	0,259179266	0,130177515	0,229679344
nli-deberta-base	14	0,274725275	0,981012658	0,194384449	0,12166405	0,216480447
nli-deberta-base	15	0,215901745	0,981012658	0,12886969	0,113553114	0,203545634
nli-deberta-base	16	0,159663866	0,993670886	0,064794816	0,10782967	0,194547708
nli-deberta-base	17	0,102133161	1	0	0,102133161	0,185337243
bge-reranker-v2-m3	1	0,850678733	0,056962025	0,940964723	0,098901099	0,072289157
bge-reranker-v2-m3	2	0,808661926	0,139240506	0,884809215	0,120879121	0,129411765
bge-reranker-v2-m3	3	0,76664512	0,221518987	0,828653708	0,128205128	0,162412993
bge-reranker-v2-m3	4	0,720749838	0,284810127	0,770338373	0,123626374	0,172413793
bge-reranker-v2-m3	5	0,673561732	0,341772152	0,711303096	0,118681319	0,176182708
bge-reranker-v2-m3	6	0,622495152	0,379746835	0,650107991	0,10989011	0,170454545
bge-reranker-v2-m3	7	0,57918552	0,455696203	0,593232541	0,113029827	0,181132075
bge-reranker-v2-m3	8	0,533290239	0,518987342	0,534917207	0,112637363	0,18510158
bge-reranker-v2-m3	9	0,486102133	0,575949367	0,475881929	0,111111111	0,186284545
bge-reranker-v2-m3	10	0,444085326	0,658227848	0,419726422	0,114285714	0,194756554
bge-reranker-v2-m3	11	0,39948287	0,727848101	0,36213103	0,114885115	0,198446937
bge-reranker-v2-m3	12	0,360051713	0,82278481	0,307415407	0,119047619	0,208
bge-reranker-v2-m3	13	0,319327731	0,911392405	0,251979842	0,121724429	0,214765101
bge-reranker-v2-m3	14	0,266968326	0,943037975	0,190064795	0,116954474	0,208100559
bge-reranker-v2-m3	15	0,210730446	0,955696203	0,125989921	0,110622711	0,198292843
bge-reranker-v2-m3	16	0,157078216	0,981012658	0,063354932	0,106456044	0,192069393
bge-reranker-v2-m3	17	0,102133161	1	0	0,102133161	0,185337243

Table E.15: Experiments with k-based cutoff

E.3 Factorial Experiments (Large Language Model)

LLM	Reasoning	Threshold	Accuracy	Recall	Specificity	Precision	F_1-score
Llama-3.3-70b	No	0	0,335488041	0,772151899	0,285817135	0,10951526	0,191823899
Llama-3.3-70b	No	1	0,396250808	0,746835443	0,35637149	0,116600791	0,201709402
Llama-3.3-70b	No	2	0,699418229	0,512658228	0,720662347	0,172707889	0,258373206
Llama-3.3-70b	No	3	0,87136393	0,202531646	0,947444204	0,304761905	0,243346008
Llama-3.3-70b	Yes	0	0,192630899	0,981012658	0,102951764	0,110635261	0,198845414
Llama-3.3-70b	Yes	1	0,400129282	0,873417722	0,346292297	0,131931166	0,22923588
Llama-3.3-70b	Yes	2	0,666451196	0,664556962	0,666666667	0,184859155	0,289256198
Llama-3.3-70b	Yes	3	0,855850032	0,221518987	0,92800576	0,259259259	0,23890785
Deep-seek-r1-70b	No	0	0,288946348	0,797468354	0,231101512	0,105527638	0,186390533
Deep-seek-r1-70b	No	1	0,351001939	0,765822785	0,303815695	0,111213235	0,194221509
Deep-seek-r1-70b	No	2	0,595992243	0,658227848	0,588912887	0,154074074	0,24969988
Deep-seek-r1-70b	No	3	0,800258565	0,411392405	0,844492441	0,231316726	0,296127563
Deep-seek-r1-70b	Yes	0	0,2708468	0,898734177	0,199424046	0,11323764	0,201133144
Deep-seek-r1-70b	Yes	1	0,363930187	0,860759494	0,307415407	0,123861566	0,21656051
Deep-seek-r1-70b	Yes	2	0,645765999	0,664556962	0,64362851	0,175	0,277044855
Deep-seek-r1-70b	Yes	3	0,804783452	0,35443038	0,856011519	0,21875	0,270531401

Table E.16: Experiments with confidence threshold

E.4 Factorial Experiments (Full)

HyDE	Reranking	Reasoning	Accuracy	Recall	Specificity	Precision	F_1-score
No	No	No	0,772462831	0,575949367	0,794816415	0,242021277	0,34082397
Yes	No	No	0,771816419	0,575949367	0,794096472	0,24137931	0,340186916
No	Yes	No	0,809954751	0,727848101	0,819294456	0,31420765	0,438931298
Yes	Yes	No	0,810601164	0,727848101	0,820014399	0,315068493	0,439770554
No	No	Yes	0,776341306	0,575949367	0,799136069	0,245945946	0,34469697
Yes	No	Yes	0,775694893	0,569620253	0,799136069	0,243902439	0,341555977
No	Yes	Yes	0,813833226	0,708860759	0,825773938	0,316384181	0,4375
Yes	Yes	Yes	0,813833226	0,708860759	0,825773938	0,316384181	0,4375

Table E.17: Experiments with confidence threshold