Enhancing Legal Compliance and Regulation Analysis with Large Language Models

by

Shabnam Hassaniahari

Thesis submitted to the
University of Ottawa
In partial fulfillment of the requirements
for the degree in
Ph.D. Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Context: Software is increasingly pervasive in regulated industries, where compliance with regulations is crucial. Driven by pressing concerns such as data protection and privacy, certain industries, including healthcare, have incorporated specific measures into their compliance frameworks to better address the role of software. Nonetheless, many industries, despite their growing reliance on digital monitoring and automation, have yet to give adequate consideration to software, as the pressure to adapt has not been as strong. This thesis focuses on two domains with complementary challenges: (i) food safety, with regulations that remain largely technology-neutral and therefore demand novel methods to connect legal provisions with systems and software requirements, and (ii) privacy and policy, where provisions already exhibit a close connection to software and systems, but compliance checking methods could be further improved.

Problem: The introduction of Industry 4.0 technologies, particularly the Internet of Things (IoT), has significantly transformed the food industry, enabling real-time monitoring and control of critical processes. Yet, food-safety regulations, like many others, remain deliberately technology-agnostic by design to ensure long-term relevance, promote innovation, and maintain market neutrality. This creates a gap between regulations and modern food-safety systems, which increasingly depend on software. Bridging this gap requires systematic identification and operationalization of requirements-related provisions within regulations. In parallel, current approaches to legal compliance checking, especially in the privacy and policy domains, often rely on sentences as the unit of analysis, apply coarse-grained classification strategies, and do not automatically provide justification for compliance decisions. These approaches also demand significant manual effort, which limits their practical usefulness for stakeholders who must demonstrate and maintain compliance.

Approach: This thesis develops and empirically evaluates a suite of methods that Large Language Models (LLMs) to address these challenges. Contributions include: (1) a Grounded Theory (GT) study of food-safety regulations, resulting in a conceptual characterization of food-safety concepts closely related to systems and software requirements; (2) an empirical evaluation of four families of LLMs (BERT, GPT, Llama, and Mixtral) for automatic classification of requirements-related provisions in food-safety regulations; (3) a study of compliance checking for privacy and policy regulations (specifically General Data Protection Regulation (GDPR) Data Processing Agreements), assessing the effectiveness of state-of-the-art LLMs (GPT, Mixtral, Mistral, Zephyr, Phi), and demonstrating the benefits of paragraph-level context and the provision of explanation and justification; and (4) a quasi-experimental study on deriving Behavior-Driven Development (BDD) artifacts from regulations using LLMs (Llama and CLaude).

Outcomes: The thesis advances regulatory analysis and compliance checking by contributing (1) a conceptual model of requirements-related food-safety concepts and the resulting annotated dataset; (2) LLM-based pipelines for classification of legal provisions and compliance checking of regulatory artifacts; and (3) empirical evidence from a quasi experiment on translating legal provisions into behavioural specifications.

Acknowledgements

I am deeply grateful to my supervisors, Prof. Mehrdad Sabetzadeh and Prof. Daniel Amyot. Their guidance, encouragement, and expertise have been instrumental in shaping this work and in my development as a researcher. I am especially thankful for their steady support and patience throughout this journey, and for their forward-looking guidance that continues to shape my next steps.

My heartfelt thanks go to my family, whose love and sacrifices have always inspired me to strive for excellence; to my partner; to my extended family in Canada for their constant encouragement and support; and to my friends, whose support has meant the world to me. Thank you all for standing by me.

This thesis was partially supported by a graduate scholarship from the University of Ottawa. Additional support was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Discovery (RGPIN-2019-06827), Discovery Accelerator (RGPAS-2019-00089), and Alliance (ALLRP-555555) programs; by the Ontario Centre of Innovation (OCI) through the Voucher for Innovation and Productivity (VIP) program; and by Stratosfy. We also gratefully acknowledge funding from Mitacs under grant number IT37879. Part of this research was conducted during an internship at New Software (formerly Knowd) under the Vector Institute's FastLane program.

Dedication

To my beloved family, whom I could not see for more than four years but whose unwavering love kept me going; to my amazing partner; and to his family and our close friends, whose constant support sustained me while I studied and worked far from home. This work is for you.

Table of Contents

List of Tables

List of Figures

List of Acronyms

AI	Artificial Intelligence
BDD	Behavior-Driven Development
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
DPA	Data Processing Agreement
FDA	Food and Drug Administration
FDR	Food and Drug Regulations
FSRG	Food-Specific Requirements and Guidance
GDPR	General Data Protection Regulation
GPT	Generative Pre-trained Transformer
GT	Grounded Theory
IoT	Internet of Things
LLM	Large Language Model
LSTM	Long Short-Term Memory
ML	Machine Learning
NLP	Natural Language Processing
RE	Requirements Engineering
RQ	Research Question
SE	Software Engineering
SFCA	Safe Food for Canadians Act
SFCR	Safe Food for Canadians Regulations
TRQ	Thesis-level Research Question

Chapter 1

Introduction

1.1 Overview and Problem Context

As software and system technologies evolve, automated processes are becoming indispensable in domains with stringent regulatory frameworks. Legal compliance, particularly in regulated industries like food safety and data privacy, requires accurate and scalable solutions to interpret, implement, and monitor adherence to regulatory standards. For example, food safety regulations demand real-time monitoring and traceability of the supply chain, while privacy and policy regulations such as the General Data Protection Regulation (GDPR) [57] emphasize data protection, accountability, and conformity.

Traditional approaches to regulatory analysis, rooted in manual interpretation and rule-based automation, struggle to scale with the increasing volume and complexity of regulations. These methods are resource-intensive, prone to inconsistencies, and lack the agility needed to adapt to jurisdictional differences or new requirements. Recent advances in Artificial Intelligence (AI) and Natural Language Processing (NLP), particularly with Large Language Models (LLMs) such as Bidirectional Encoder Representations from Transformers (BERT) [52] and Generative Pre-trained Transformer (GPT) [39], have opened new possibilities for automating tasks at the intersection of Requirements Engineering (RE) and legal compliance. By leveraging LLMs' capabilities in understanding and generating natural language, this thesis aims to address the critical gaps in current automation methodologies.

1.2 Motivation

Regulated industries, such as food safety and data privacy, increasingly rely on software-intensive systems for compliance management. Yet the challenges they face differ in scope and nature. In food safety, the core difficulty lies in the technology-agnostic design of regulations, which offer little direct guidance for translating provisions into software-relevant requirements. In contrast, privacy and policy regulations such as the GDPR are already

software-centric, but current compliance approaches remain highly manual, fragmented, and lacking in scalability or explainability. These two domains, though different in their challenges, both highlight the limitations of existing methods for connecting regulatory frameworks with modern software systems.

Across both domains, three challenges emerge in operationalizing regulations within software systems.(1) Regulations such as the Codex Alimentarius [86] and the Safe Food for Canadians Regulations (SFCR) [83], are vast and intricate, often written without explicit references to modern systems. Translating provisions like temporal constraints, traceability records, and measurement constraints into software artifacts remains a manual, time-consuming, and error-prone process. (2) Many compliance-checking approaches rely on sentence-level analysis or rule-based systems, which fail to scale with the increasing volume of legal content. These methods lack the ability to interpret broader relationships, such as hierarchical definitions, cross-references, and multi-paragraph dependencies, which are essential for accurate assessment. (3) Existing tools often focus on classification and information extraction but provide little to no justification for their decisions. In high-stakes sectors like data privacy and policy, the lack of explainability undermines trust.

Addressing these challenges calls for pipelines that scale efficiently, reason over broader textual scopes, and provide transparent justifications, while producing software artifacts that are accessible to software engineers and testers. In light of these issues, this thesis investigates the application of LLMs to bridge the gap between regulatory frameworks and modern software systems. This thesis delivers its contributions through three interconnected studies:

1. **Classifying Requirements-Related Provisions in Food-Safety Regulations:** Regulations like the SFCR and the Codex Alimentarius create significant barriers to identifying software-relevant provisions that can inform software development for monitoring food-monitoring software and systems. The first study in this thesis addresses this gap by leveraging LLMs to classify requirements-related provisions in food-safety regulations. By outperforming traditional keyword-based and Long Short-Term Memory (LSTM) methods in terms of accuracy, this work demonstrates how LLMs can serve as foundational models for automating compliance workflows.

2. **Improving Compliance Automation in Privacy and Policy Regulations:** Building on the first study, the second study broadens the focus to privacy and policy regulations, such as the General Data Protection Regulation (GDPR) and Data Processing Agreements (DPAs). Existing compliance automation methods in this domain are often constrained to sentence-level analysis and lack justifications for compliance decisions, limiting their transparency.This study highlights how LLMs can analyze regulatory texts at paragraph levels, enabling more context-aware interpretations. By enhancing accuracy and offering transparency, this work underscores the potential of LLMs to reimagine compliance automation in highly regulated sectors.

3. **Deriving Behavioural Specifications from legal texts:** Finally, the third study ties these insights together by addressing practical challenges when it comes to im-

plementation and testing, moving from legal texts to software artifacts, particularly the divide between technical and non-technical stakeholders. It introduces a novel process for translating legal texts into structured Gherkin specifications [156] using LLMs. Behavior-Driven Development (BDD) [203] makes legal texts more accessible to software engineers. The integration of behavioural specifications with LLM capabilities bridges the gap between technical and non-technical stakeholders.

1.3 Research Questions

Three thesis-level research questions are identified here. As each of the chapters answering these questions (Chapters 3-5) has its own local Research Questions (RQs), the thesis-level ones, Thesis-level Research Questions (TRQs), are labelled TRQ1 to TRQ3 to avoid confusion. These questions reflect the technical and practical challenges addressed in this thesis while aligning with its goal of advancing compliance automation.

TRQ1: How can Grounded Theory (GT) and LLMs be used to identify and classify requirements-related provisions in underexplored regulatory domains, such as food safety regulations? This question explores the use of GT to establish domain-relevant regulatory concepts and leveraging LLMs for their automated classification capabilities.

TRQ2: How does expanding LLM analysis to broader textual scopes (e.g., paragraphs) improve legal compliance automation in domains such as data privacy regulations? This question captures the exploration of contextual analysis to improve the precision of compliance automation.

TRQ3: How can LLMs be leveraged to improve the automation of the derivation of behavioural specifications from regulatory texts? This question is focused on BDD specifications, specifically Gherkin, and their role in facilitating collaboration among stakeholders.

1.4 Contributions

This thesis integrates findings from several core studies. These publications, co-authored with co-supervisors and an industry partner, include:

1. **Shabnam Hassani**, Mehrdad Sabetzadeh, and Daniel Amyot "An Empirical Study on LLM-based Classification of Requirements-related Provisions in Food-safety Regulations". *Empirical Software Engineering*, 2025. [107]

    - This paper, covered in Chapter 3, addresses TRQ1.

2. **Shabnam Hassani**, Mehrdad Sabetzadeh, Daniel Amyot, and Jian Liao "Rethinking Legal Compliance Automation: Opportunities with Large Language Models". *IEEE International Requirements Engineering Conference (RE)*, RE@Next track, 2024. [109]

- This paper, covered in Chapter 4, addresses TRQ2.

3. **Shabnam Hassani**, Mehrdad Sabetzadeh, and Daniel Amyot "From Law to Gherkin: A Human-Centred Quasi-Experiment on the Quality of LLM-Generated Behavioural Specifications from Food-Safety Regulations", submitted to *Information and Software Technology*, 2025. [108]

   - This paper, covered in Chapter 5, addresses TRQ3.

4. **Shabnam Hassani** "Enhancing Legal Compliance and Regulation Analysis with Large Language Models". *IEEE International Requirements Engineering Conference (RE)*, Doctoral Symposium, 2024. [96]

   - This short paper provides an overview of the thesis itself

The next three subsections enumerate the expected contributions for each of the thesis research questions (TRQs).

1.4.1 Characterizing Food-Safety Regulations for Software-relevant Requirements Derivation (TRQ1)

Food-safety regulations are deliberately technology-neutral. While this design ensures long-term applicability, it creates challenges for software-intensive food-safety systems that must operationalize regulatory provisions. Chapter 3 addresses this gap by introducing a content model, derived through a GT study of Canadian food-safety regulations, that classifies provisions based on their relevance to systems and software requirements. The resulting framework provides a systematic basis for filtering and organizing legal texts into system-relevant categories.

The classification framework is further shown to generalize across jurisdictions, as models trained on Canadian provisions perform consistently on U.S. regulations. This cross-jurisdictional robustness highlights the potential of the approach for broader adoption in regulatory contexts beyond Canada.

The chapter also investigates the use of LLMs for automating the classification of requirements-related provisions. It demonstrates how LLMs substantially outperform baselines such as Bidirectional Long Short-Term Memory (BiLSTM) and keyword search, offering accuracy gains, scalability, and adaptability even in the presence of sparse annotations.

A key contribution is the demonstration that only a small fraction of regulatory provisions, approximately one quarter, directly relate to systems and software requirements. Automating the identification of this subset significantly reduces manual effort while maintaining high fidelity, underscoring the importance of automated filtering for practical compliance analysis.

The proposed pipeline further supports regulatory stakeholders through metadata-driven guidance. Food-safety regulations often associate requirements with food-type labels, such as "meat" or "dairy." Rather than requiring stakeholders to define compliance

concepts separately for each food type, the framework demonstrates that general regulatory concepts (e.g., "temperature," "mass") can be systematically combined with food-type labels, forming a Cartesian product. This mechanism eliminates repetitive, domain-specific classifications while ensuring comprehensive coverage of requirements. By relying on general concepts and leveraging the Cartesian product, stakeholders can navigate the regulatory space more effectively, identify actionable provisions, and establish traceability links between legal requirements and software design elements. The resulting guidance enhances transparency, reduces complexity, and ensures scalability in compliance workflows across diverse food categories. This positions metadata-driven guidance as a key enabler of efficient and scalable compliance management.

1.4.2 Advancing Compliance Automation with Broader Contextual Analysis (TRQ2)

Chapter 4 investigates how compliance automation can be advanced by moving beyond sentence-level analysis and by providing explanations for compliance decisions. Existing approaches are hindered by three main limitations: (1) a predominant reliance on sentences as units of analysis, (2) the absence of automatically generated justifications for decisions, and (3) rule-evaluation strategies that are either overly coarse-grained or require significant manual effort.

This study introduces a shift from sentence-level to paragraph-level analysis of legal texts using LLMs. Paragraph-level analysis captures broader context and cross-references, such as definitions and cumulative provisions, that are essential for accurate compliance assessments, but are often missed when processing sentences in isolation. Leveraging the extended context windows of modern generative LLMs enables more nuanced and reliable compliance evaluations.

A further contribution of this study is the integration of justification mechanisms into compliance checking. Through carefully constructed prompts that incorporate compliance rules, contextual passages, and rule-specific questions, the approach elicits natural-language justifications alongside compliance decisions. These justifications enhance the transparency and accountability of automated legal assessments, addressing a long-standing concern in compliance automation.

Finally, the study evaluates the practical feasibility of paragraph-level analysis at scale. Empirical evidence shows that the approach processes thousands of regulatory paragraphs per hour with sub-second latency per paragraph. This level of efficiency demonstrates the viability of the method for real-world compliance tasks, reducing both manual effort and associated costs for stakeholders.

1.4.3 Deriving Behavior-Driven Development Specifications (TRQ3)

Chapter 5 explores the automation of translating legal provisions into behavioural specifications by integrating LLMs with BDD. This chapter adopts Gherkin, the de-facto syntax for

BDD, as our target format. Gherkin's Given–When–Then structure makes requirements understandable to both technical and non-technical stakeholders, while also being directly reusable for test automation. This choice ensures that provisions from food-safety regulations can be operationalized in a structured, developer-friendly manner, thus bridging a long-standing gap between abstract legal texts and concrete software artifacts.

The study addresses the practical challenge of generating high-quality Gherkin specifications from dense, technology-neutral legal texts. While prior research has applied LLMs to user stories or informal requirements, the use of LLMs to derive legal compliance-oriented BDD artifacts has not been systematically studied. The focus is on food-safety regulations as a representative domain, leveraging their complex but widely applicable provisions.

This chapter presents the first human-subject quasi-experiment to evaluate the quality of LLM-generated Gherkin specifications from legal texts. Ten participants rated 60 generated specifications across five quality criteria: Relevance, Completeness, Clarity, Singularity, and Time Savings and provided qualitative feedback.

The results highlight both strengths and limitations. Participants consistently rated the specifications highly across all five criteria. However, recurring issues were identified, including omissions of clauses, occasional hallucinations, and violations of BDD's singularity principle. Categorizing these issues provides actionable insights for improving both LLM prompting strategies and downstream human-in-the-loop refinement processes.

1.5 Limitations

This thesis explores the use of LLMs for compliance automation and regulatory analysis, with a deliberate focus on domains such as food safety and privacy and policies. While the proposed pipelines demonstrate significant potential, several limitations must be acknowledged.

TRQ1. The classification framework and datasets were built primarily on Canadian and U.S. food-safety regulations. Although cross-jurisdictional robustness was observed, extending the framework to other jurisdictions remains untested. The GT analysis reflects the researcher's interpretation, and LLM classification, though superior to baselines, may still underperform with less structured or low-resource legal texts.

TRQ2. The shift from sentence- to paragraph-level analysis improves contextual accuracy, but does not fully capture dependencies across multiple sections or documents. Also, justifications generated by LLMs need to be assessed for legal rigor in professional settings. Moreover, the evaluation is limited and leaves the scalability of other legal corpora uncertain. Current LLMs also remain challenged by multi-step reasoning, especially when obligations span cross-references or nested provisions.

TRQ3. The quasi-experimental evaluation involved 10 participants and 60 specifications. While informative, this is a modest sample size, and participants were technically trained students rather than professional regulators or industry practitioners. The study focused exclusively on food-safety provisions, limiting generalizability to other domains.

Recurring issues such as omissions, hallucinations, and violations of BDD's singularity principle indicate that human validation remains indispensable.

Cross-cutting limitations. Several broader constraints affect all three studies. First, legal texts often contain intricate cross-references, hierarchies, and implicit assumptions that LLMs struggle to capture reliably. Second, LLMs remain prone to hallucinations and lack robust reasoning for complex compliance tasks. Third, scalability is constrained by the computational cost of large-scale fine-tuning, particularly for resource-limited organizations. Finally, the regulatory landscape is dynamic; ensuring models remain aligned with evolving provisions requires continuous updates, and this thesis does not address versioning challenges.

1.6 Thesis Outline

Chapter 2 provides an overview of the fundamental concepts that are relevant to the thesis.

Chapter 3 presents the published paper "An Empirical Study on LLM-based Classification of Requirements-related Provisions in Food-safety Regulations", which addresses TRQ1.

Chapter 4 presents a reformatted (but otherwise unchanged) version of the published paper "Rethinking Legal Compliance Automation: Opportunities with Large Language Models", which addresses TRQ2.

Chapter 5 presents the submitted paper "From Law to Gherkin: A Human-Centred Quasi-Experiment on the Quality of LLM-Generated Behavioural Specifications from Food-Safety Regulations", which addresses TRQ3.

Finally, Chapter 6 concludes the work presented in this thesis and identifies important future work directions.

Chapter 2

Background

Chapter overview. This establishes the background and defines the concepts that the thesis relies on. Detailed discussions and analyses of these concepts are presented in subsequent chapters.

2.1 Background

The aim here is sufficiency rather than exhaustiveness. For each concept, the emphasis is on the pieces used later:

2.1.1 Food-safety Regulations

Food-safety regulations protect public health by setting standards for the production, processing, packaging, distribution, and record-keeping of food products across the supply chain. In Canada, the Safe Food for Canadians Act (SFCA) and its accompanying regulations, Safe Food for Canadians Regulations (SFCR) [83], unify multiple federal regimes into a streamlined regulatory framework aligned with Codex Alimentarius [86].

SFCR is organized around several pillars:

1. Licensing: Establishing when and under what conditions, a Safe Food for Canadians license is required.
2. Preventive controls: Requirements for hazard identification, written preventive control plans, sanitation, and verification.
3. Traceability: Obligations to track and maintain records in the food supply chain.
4. Packaging, labelling, and grades: Mandatory information and quality standards for consumer protection.

In addition, SFCR defers many commodity-specific requirements to Food-Specific Requirements and Guidance (FSRG) [81]. FSRG specifies product-dependent obligations, e.g., pasteurization temperatures for dairy, inspection regimes for meat, and grading standards for produce. Also, cross-references tie SFCR to its enabling Act (SFCA), as well as to the Food and Drugs Act (FDA, Canada) and the Food and Drug Regulations (FDR).

Technology-neutral language in these regulations ensures adaptability but creates a gap between legal texts and the modern systems used for compliance, such as Internet of Things (IoT) enabled monitoring solutions, which are increasingly used for compliance and monitoring tasks [33]. This thesis leverages LLMs to bridge this gap by extracting and classifying regulatory provisions and identifying their implications for systems and software requirements.

In Chapter 3, a systematic GT study is conducted and a conceptualization of software-relevant provisions in food safety regulations in Canada [83] is provided. Software-relevant concepts based on our GT study include: *Sensing and monitoring*: obligations to monitor, e.g., temperature, humidity, weight, PH; *Time constraints*: temporal conditions such as maximum durations, intervals, or delays; and *Data*: requirements for identification, storage, format, data provenance, lineage, lot tracking, and supply chain visibility. These concepts offer a precise lens for interpreting food safety legal texts in terms of their relevance to software and systems. Furthermore, by aligning legal provisions with these categories in Chapter 5, provisions are operationalized into actionable behavioural software artifacts, specifically Gherkin.

2.1.2 Grounded Theory (GT)

Grounded Theory is a qualitative research methodology aiming to generate theory through systematic data collection, coding, memoing, and constant comparison [76]. Rather than starting from a priori hypotheses, GT proceeds iteratively, letting explanatory categories and their relationships emerge from close engagement with the corpus. In Software Engineering (SE), GT has been widely used to build theory about practices, phenomena, and artifacts that are insufficiently theorized [207]. Contemporary GT work typically follows one of three families: Glaserian (classic), Straussian, and Constructivist [42, 75, 208].

Across GT variants, four procedure families recur [164, 207, 208]:

1. Theoretical Sampling: evolving selection of data sources to elaborate or challenge emerging categories until theoretical saturation (no new properties nor relationships) is reached.

2. Coding: a staged analysis where data is systematically fractured and recombined into categories. While all GT variants employ staged coding, the precise terminology differs: Glaser [75] distinguishes substantive and theoretical coding, Strauss and Corbin [208] propose open, axial, and selective coding, and Charmaz [42] advocates initial and focused coding.

3. Constant Comparison: continual juxtaposition of incidents, codes, and categories to sharpen boundaries, discover negative cases, and ensure analytic fit.

4. Memoing: analytic notes capturing hypotheses about category properties, dimensional ranges, boundaries, and relationships; memos form the backbone of the emerging theory and its audit trail.

The product of GT is a set of categories with defined properties and theoretically significant relationships that explain the variation in the phenomenon studied.

GT is particularly suited for exploring under-defined domains, such as the intersection of food-safety regulations and system/software requirements. In this thesis, the GT product is (i) a taxonomy of software-relevant concepts in food-safety regulations and (ii) relationships that represent the hierarchy of concepts. This thesis adopts the *Straussian* variant of GT in Chapter 3, which allows engagement with the existing literature to refine conceptual categories and relationships [208].

2.1.3 Large Language Models (LLMs)

Large Language Models, such as BERT and GPT, are advanced neural networks trained on vast textual datasets to understand and generate human-like text [52, 185]. Modern LLMs are predominantly based on the transformer architecture [224], which relies on self-attention to capture long-range dependencies and contextual relationships more efficiently than earlier recurrent and convolutional approaches. Transformers consist of stacked self-attention and feed-forward layers with positional encodings to retain order information [224]. Two pretraining objectives dominate:

- Causal (autoregressive) modeling: predicts the next token x_t given previous tokens $x_{<t}$, optimizing $p(x_t \mid x_{<t})$. This objective favors fluent text generation and supports in-context learning [39, 185].

- Masked language modeling: reconstructs masked tokens given bidirectional context, optimizing $p(x_{\text{mask}} \mid x_{\setminus\text{mask}})$. This objective yields strong representations for classification, retrieval, and sequence labelling [52].

From these two pretraining paradigms, LLMs are often characterized as either: discriminative models (e.g., BERT), which excel at classification and extraction tasks by producing contextual embeddings optimized for downstream prediction, or generative models (e.g., GPT), which are optimized for sequence continuation and can produce coherent free-form text, enabling tasks such as summarization, translation, and structured requirement generation.

LLMs therefore excel in text classification, summarization, and comprehension tasks, making them highly effective for processing complex legal and regulatory documents. By fine-tuning, few-shot learning, and zero-shot learning experiments with both discriminative and generative LLMs on annotated datasets derived from food-safety regulations, this thesis demonstrates their utility in automating compliance tasks, significantly outperforming traditional methods not based on transformer models [23, 90].

2.1.4 General Data Protection Regulation (GDPR)

The General Data Protection Regulation [57] is the flagship privacy law of the European Union, comprising 173 recitals and 99 articles across 11 chapters [114]. GDPR governs the collection, processing, storage, and transfer of personal data within the EU, while also applying extraterritorially to organizations outside the EU that target or monitor EU residents.

GDPR distinguishes between two principal roles: *data controllers*, who determine the purposes and means of processing personal data, and *data processors*, who act on behalf of controllers and must follow their instructions. Controllers face stricter obligations, including adherence to six core principles (fairness, purpose limitation, data minimization, accuracy, storage limitation, and security), as well as ensuring data subject rights such as access, erasure, and portability. Processors must implement technical and organizational safeguards, maintain records of processing, and report data breaches without undue delay [127].

A central regulatory instrument mandated by GDPR is the Data Processing Agreements (DPAs), outlined in Article 28. DPAs are legally binding contracts that specify the responsibilities of controllers and processors regarding personal data. They must include details such as the duration of processing, categories of data subjects, obligations for deletion or return of data upon termination, and provisions for international data transfers [13]. As DPAs explicitly structure software-relevant obligations, for example, ensuring secure storage, deletion protocols, or access controls, they are highly relevant to the Requirements Engineering (RE) community. Several prior RE works on compliance automation have focused on mapping DPA content to GDPR rules [13, 127].

Our work builds on insights from these studies by exploring how paragraph-level context and generative LLMs can improve compliance verification. In this thesis, GDPR and its associated DPAs serve as a testbed for assessing the adaptability of LLMs to compliance analysis tasks in privacy and data protection law.

2.1.5 Behavior-Driven Development (BDD)

Behaviour-Driven Development [203] is an Agile requirements and testing methodology that emphasizes shared understanding between technical and non-technical stakeholders. BDD builds on Test-Driven Development but introduces a ubiquitous language that bridges the gap between business intent and executable test cases. System behaviours are specified as scenarios expressed in structured natural language, typically in the format "Given–When–Then". This representation helps ensure that requirements are not only verifiable but also comprehensible to stakeholders without programming expertise.

In practice, BDD supports requirements validation by enabling stakeholders to collaboratively define acceptance criteria before implementation begins. These criteria double as automated test cases when executed through supporting frameworks. Within RE, BDD has been recognized as a powerful vehicle for creating living documentation that stays synchronized with system behaviour throughout development.

In this thesis, BDD is applied to the translation of legal provisions, such as those in food safety legal provisions, into actionable behaviour specifications, specifically Gherkin. Chapter 5 evaluates how LLMs can assist in automatically generating BDD scenarios from legal texts, investigating both the potential of this automation and its current limitations.

2.1.6 Gherkin

Gherkin is the domain-specific language most commonly used to implement BDD scenarios. It provides a structured syntax composed of keywords such as Feature, Scenario, Given, When, Then, And, and But [156]. This syntax ensures that scenarios remain both human-readable and machine-processable. Importantly, Gherkin feature files can be directly executed using frameworks such as Cucumber [111], SpecFlow [211], and JBehave [130], thereby binding natural-language specifications to underlying code implementations. From an RE perspective, Gherkin supports the creation of executable acceptance criteria that reduce ambiguity and enable traceability between requirements and tests.

As Gherkin is structured yet flexible, it serves as a target formalism for translating legal provisions into testable artifacts. For instance, a GDPR obligation that "data shall be erased without undue delay upon request" can be captured as a Gherkin scenario specifying preconditions (*Given a user submits a deletion request*), actions (*When the request is processed*), and expected outcomes (*Then the system must erase the data within a specified timeframe*).

In this thesis, Gherkin serves as the target language for automatically generated specifications from legal texts. The choice of Gherkin reflects both its industrial adoption and its suitability for bridging natural-language legal texts and automated compliance verification and testing.

2.2 Summary

This chapter consolidated the background used throughout the thesis. It clarified the aspects of food-safety regulations, GT, LLMs, GDPR, BDD, and Gherkin with the level of detail needed for later chapters.

The following chapters discuss three TRQs: Chapter 3 (TRQ1) derives a GT-based content model for food-safety legal texts and evaluates LLMs for classifying requirements-related provisions; Chapter 4 (TRQ2) scales compliance analysis to paragraph context with decision justifications; Chapter 5 (TRQ3) translates legal texts into BDD specifications, specifically Gherkin, and assesses quality with human subjects; Chapter 6 synthesizes the key findings across TRQs.

Chapter 3

An Empirical Study on LLM-based Classification of Requirements-related Provisions in Food-safety Regulations

As Industry 4.0 transforms the food industry, the role of software in achieving compliance with food-safety regulations is becoming increasingly critical. Food-safety regulations, like those in many legal domains, have largely been articulated in a technology-independent manner to ensure their longevity and broad applicability. However, this approach leaves a gap between the regulations and the modern systems and software increasingly used to implement them. In this article, we pursue two main goals. First, we conduct a Grounded Theory study of food-safety regulations and develop a conceptual characterization of food-safety concepts that closely relate to systems and software requirements. Second, we examine the effectiveness of two families of large language models (LLMs) – BERT and GPT –in automatically classifying legal provisions based on requirements-related food-safety concepts. Our results show that: (a) when fine-tuned, the accuracy differences between the best-performing models in the BERT and GPT families are relatively small. Nevertheless, the most powerful model in our experiments, GPT-4o, still achieves the highest accuracy, with an average *Precision* of 89% and an average *Recall* of 87%; (b) few-shot learning with GPT-4o increases *Recall* to 97% but decreases *Precision* to 65%, suggesting a trade-off between fine-tuning and few-shot learning; (c) despite our training examples being drawn exclusively from Canadian regulations, LLM-based classification performs consistently well on test provisions from the US, indicating a degree of generalizability across regulatory jurisdictions; and (d) for our classification task, LLMs significantly outperform simpler baselines constructed using LSTM networks and automatic keyword extraction.

3.1 Introduction

Software is increasingly pervasive in regulated industries where compliance with legal provisions is a must. Examples of regulated industries include healthcare, transportation,

energy, food and agriculture. Driven by pressing concerns such as data protection and privacy, some industries, such as healthcare, have incorporated specific measures into their compliance frameworks to better address the role of software [36]. Nonetheless, many industries, where the pressure to adapt has not been as strong, have yet to give adequate consideration to software. Our work in this chapter is focused on one of these industries, namely the *food industry*.

The most critical regulatory aspect for food is *food safety*. Food safety refers to the measures taken to ensure that food products are free from harmful contaminants and that they are safe for human consumption [234]. Food-safety regulations set standards and guidelines for the production, processing, packaging, labelling, storage, and distribution of food products. Food safety is arguably the most far-reaching safety concern, as it affects the life of every individual. This makes food-safety regulations among the most impactful and significant in terms of their importance. An estimated 600 million people – nearly 1 in 13 – fall ill each year from contaminated food, with 420,000 fatalities, disproportionately affecting children under five. Economically, unsafe food results in an estimated annual loss of $110 billion USD in productivity and medical expenses in low- and middle-income countries, placing a significant strain on healthcare systems and impeding socioeconomic development [234]. To address food-safety risks, there is a robust and growing market for food-safety testing and monitoring, the size of which is projected to be around $24 billion USD by the end of 2024 [189].

In recent years, Industry 4.0 [74] has significantly transformed the food industry. One of the main thrusts of this transformation has been the introduction of the IoT into food-supply chains. This has enabled companies to monitor and control various processes, including those related to food safety, in real-time [33]. For example, food-service operators can now use sensors to monitor environmental conditions such as temperature and humidity during food storage and transportation to ensure that food products are maintained at optimal conditions.

Food-safety regulations, like many other types of legal measures, are technology-agnostic and are designed to ensure long-term relevance, promote innovation, and maintain market neutrality. While this flexibility is beneficial, it also creates a gap between regulations and modern food-safety systems, which increasingly rely on software. Closing this gap requires developing techniques that enable a smooth transition from regulations to systems and software requirements.

Motivation. Our research was prompted by a stated need from a collaborating industry partner, Stratosfy (`https://stratosfy.io`). Stratosfy develops an IoT-based temperature monitoring system for food products. This system helps food businesses such as restaurants and supermarkets avoid spoilage by monitoring their cooling devices (e.g., walk-in coolers, reach-in coolers and milk dispensers), and sending warnings if anomalies happen in temperature patterns. A frequent request from the clients of this system is to provide additional features that would facilitate demonstration of compliance with food-safety regulations. The partner is thus interested in transforming food-safety provisions into feature requirements in line with the capabilities afforded by the existing IoT technologies.

Identifying which food-safety provisions are relevant to systems and software require-

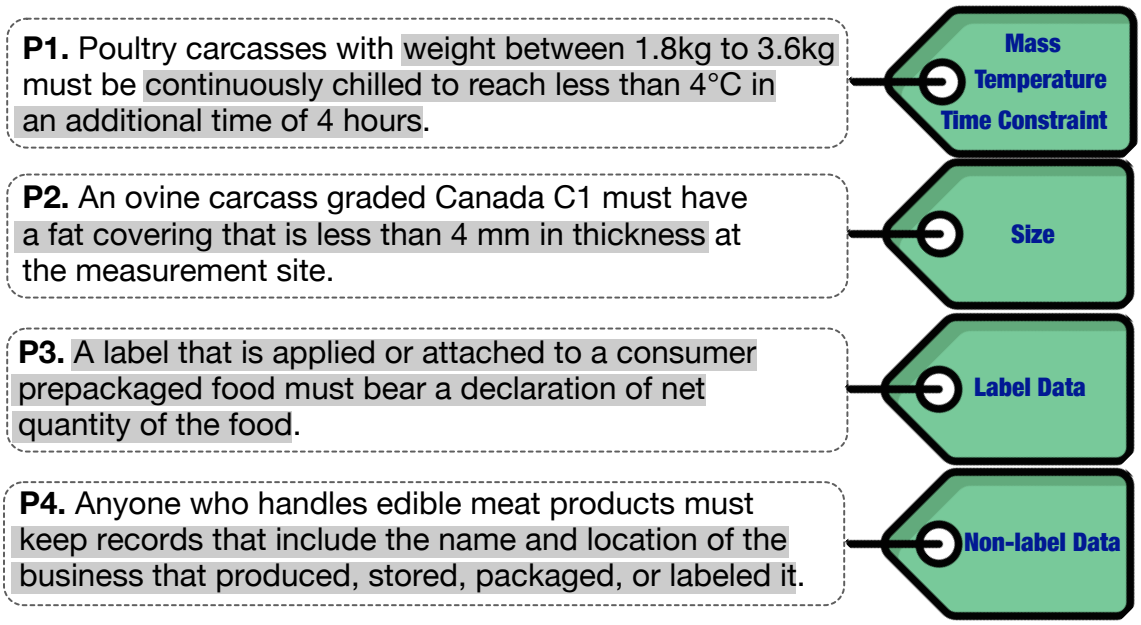


Figure 3.1: Example provisions, P1–P4, from the Safe Food for Canadians Regulations [83] alongside labels that classify the requirements-related content of these provisions.

ments is not a straightforward task, since food-safety regulations have not been written with systems and software in mind. The *first challenge* that arises here is characterizing the food-safety concepts that are most relevant to automated food-safety systems.

A *second challenge* is posed by the sheer volume of text in food-safety regulations. Even with a solid understanding of the relevant concepts and expertise in legal language, manually searching for instances of these concepts within vast amounts of legal content – often across multiple jurisdictions – can be an overwhelming task. A prime example of the scale involved is the Codex Alimentarius (Latin for "Food Code"), developed jointly by the Food and Agriculture Organization (FAO) and World Health Organization (WHO). Established in 1963, the Codex is an intentionally recognized compilation of standards, guidelines, and codes of practice aimed at ensuring the safety, quality, and fairness of international food trade. As of this writing, the Codex has 87 guidelines, 237 commodity standards, and 57 codes of practice, totaling *thousands* of pages on material related to food safety. Individual countries adapt this material into their national regulatory frameworks, often tailoring them to address local public health concerns, consumer preferences, and trade priorities. This process frequently results in the creation of extensive regulatory corpora, which can amount to thousands of pages for each country [87–89]. When dealing with a large volume of legal documents, automated assistance can substantially improve efficiency and accuracy in identifying provisions that are requirements-related.

Objectives. The objectives of this chapter are two-fold: (1) to provide a characterization of regulatory concepts that are relevant to the development of food-safety systems, and (2) to develop an automated pipeline to accurately identify instances of these concepts in an input legal text.

We illustrate our objectives using the example of Fig. 3.1. The provisions P1–P4 in this figure originate from the SFCR [83], discussed further in Section 3.2. These have been modified from their original form to simplify illustration. The original provisions are available in our online repository [101] and Appendix A.

In our example, P1 could imply sensing requirements for the measurement of *Mass* and *Temperature*. Furthermore, P1 is time-sensitive, indicating a *Time Constraint* that can potentially impact monitoring requirements. P2 may imply additional sensing requirements, say if digital calipers are used to measure fat thickness (*Size*). P3 specifies information that food *Label*s must include, with potential implications for sensing, monitoring and data management. P4 envisages important traceability information *beyond* food-product labels (i.e., *Non-label Data*) that may need to be collected during operation.

Although none of the provisions in Fig. 3.1 directly refer to automation, the provisions can induce different systems and software requirements depending on the nature and extent of automation implemented in a proposed food-safety system. Our first objective has to do with defining what labels (classes), e.g., *Temperature* and *Time Constraint*, to classify provisions by. Given a food-safety-related legal text, our second objective is concerned with automatically attaching labels to the provisions in the text, as exemplified in Fig. 3.1. We are particularly interested in *LLMs* as enablers of such automation. These models are capable of understanding, generating, and interpreting natural language in context, thus offering the potential to more accurately identify and classify relevant regulatory concepts.

Research Questions (RQs):

We achieve our objectives by answering the following RQs:

- **RQ1: What concepts in food-safety regulations are most relevant to systems and software requirements?** We answer RQ1 by conducting a GT study, focusing on Canadian food-safety regulations.
- **RQ2: How accurately can LLMs classify requirements-related provisions in food-safety regulations?** We answer RQ2 by systematically examining classification models built using variants from two LLM families: BERT and GPT.
- **RQ3: How do LLM-based classification models compare to simpler baselines in the context of food-safety regulations?** We answer RQ3 by comparing the best-performing LLM-based classification models identified in RQ2 with two baseline methods: one using a long short-term memory (LSTM) model and the other based on keyword search.

Contributions. The contributions of this chapter are as follows:

1. *A GT study of Canadian food-safety regulations.* Using a GT study, we systematically analyze federal food-safety regulations in Canada to identify key concepts that influence food-safety system requirements. We focus on Canadian regulations because they are well-established and closely aligned with the operational domain of our industry partner.

2. *An empirical evaluation of LLMs for classifying food-safety provisions.* We create an automated pipeline that can be instantiated with different LLMs for classifying the content of legal texts related to food safety. We use the information elements from the content model developed in our GT study (i.e., our first contribution) as classification labels. We consider state-of-the-art variants (as of this writing) from four families of LLMs: BERT, GPT, Llama, and Mixtral. During our preliminary investigation, we exclude Llama and Mixtral due to technical limitations, most notably their inability to follow instructions in our application context. We then conduct a detailed empirical evaluation of the variants in the remaining two families: BERT and GPT. Our results show that: (a) when fine-tuned, the accuracy differences between the best-performing models in the BERT and GPT families are relatively small. Nevertheless, the most powerful model in our experiments, GPT-4o, still has an edge with an average *Precision* of 89% and an average *Recall* of 87%; (b) few-shot learning with GPT-4o increases *Recall* to 97% but decreases *Precision* to 65%, suggesting a trade-off between fine-tuning and few-shot learning; (c) despite our training examples being drawn exclusively from Canadian regulations, LLM-based classification performs consistently well on test provisions from the US, indicating a degree of generalizability across regulatory jurisdictions; and (d) for our classification task, LLMs significantly outperform simpler baselines constructed using long short-term memory (LSTM) networks and automatic keyword extraction.

Target Users and Practical Applications. Our work aims to establish a conceptual basis for the development of regulatory requirements for food-safety systems and to systematically assess the readiness of LLMs as an assistive technology in this context. While acknowledging that our conceptualization of requirements-related concepts in food-safety regulations and our automated support for identifying these concepts have not yet been validated in industrial applications, our contributions target the following stakeholders as the main beneficiaries: suppliers and integrators of food-safety monitoring systems, who are responsible for ensuring compliance with relevant regulations. To assist these stakeholders, our work paves the way for implementing two important use cases:

1. **Use Case 1 (Filtering Irrelevant Content)**. One major advantage of automated classification is providing the ability to filter out irrelevant content. Our GT study indicates that only about 25% of the paragraphs in SFCR documents are relevant, meaning that they include occurrences of requirements-related concepts. By developing an accurate automated content classifier, we provide an effective tool to filter out content unrelated to systems and software requirements. This in turn enables limited human resources to focus on content that is more likely to be relevant and have a significant impact on requirements.

2. **Use Case 2 (Metadata-driven Guidance)**. Regulations often contain complex legal jargon, increasing the likelihood that requirements-relevant content may be overlooked if reviewed without guidance. To address this issue, an additional layer of *metadata* is useful for highlighting how specific provisions relate to systems and software, such as by requiring some kind of sensing, measurement, or data collection. As shown by our illustrative example in Fig. 3.1, classification labels help clarify

the nature of each provision's relevance to systems and software requirements. This in turn enables stakeholders to focus more effectively on the relevant aspects of the regulations.

Novelty. Compliance with legal provisions is a key concern for software-intensive systems. This has led the Requirements Engineering (RE) community to develop automation for classifying and extracting requirements-related information from legal texts. However, most previous research in this direction, e.g., from [239, 240] and [202], has focused on high-level concepts such as those in the Hohfeldian system [40] or deontic logic [135]. While such research is useful for domains where a clear conceptual link exists between law and software-intensive systems, it is insufficient for domains like food safety, where no such link has been established yet. The RE literature further acknowledges the importance of domain-focused classification of legal concepts [36, 73]. Yet, most current studies address privacy regulations [12, 29, 58, 213, 214, 238], which are relatively new and align closely with software systems. The study of traditional domains whose regulatory frameworks were not designed with such systems in mind remains limited. Our work is the first to attempt to address this gap for the domain of food safety.

Structure. Section 3.2 discusses background. Section 3.4 presents our GT study on Canadian food-safety regulations. Section 3.5 describes our pipeline for automated text processing and classification. Section 3.6 reports on the accuracy of this pipeline over food-safety regulations, experimenting with model variants from different LLM families. Section 3.7 outlines key practical implications. Section 3.3 compares with related work. Section 3.9 provides concluding remarks. Section 3.8 directs to our replication package.

3.2 Background

This section presents the necessary background on food-safety regulations and on the enabling AI technologies used.

3.2.1 Food-safety Regulations

To ensure that the food supply is safe, countries around the world have established regulatory frameworks to govern the production, distribution, and consumption of food products [234]. Our primary focus is on the Safe Food for Canadians Regulations (SFCR) [83]. SFCR consolidates several federal Canadian food regulations that apply, among other things, to the import, export, and inter-provincial trade of food, including ingredients [84]. SFCR sets out both general food requirements as well as specific requirements for individual food products. In many cases, SFCR defers the elaboration of requirements for specific food products to FSRG regulations [81]. The FSRG regulations notably cover dairy products, egg products, fish, fruits and vegetables, meat, honey, manufactured products, and maple. We use content from FSRG regulations both in our GT study (Section 3.4) and in our evaluation (Section 3.6).

In addition to SFCR and the associated FSRG regulations, we consider in our work selected parts of the food-safety regulations by the US Food and Drug Administration (FDA) [85]. FDA regulates various aspects of food production and handling, including labelling, ingredients and packaging. We use FDA regulations to examine the extent to which our classification solution is transferable to jurisdictions outside Canada.

3.2.2 Grounded Theory (GT)

To identify food-safety concepts that are relevant to systems and software requirements, we apply GT. GT is a qualitative research method originally described by [76]. The primary objective of GT is to develop theory through the systematic collection and analysis of data, rather than to test or validate pre-existing theories. Noting that, to the best of our knowledge, no systematic conceptualization (theory) of food-safety concepts related to systems and software requirements has been previously established, GT offers an effective method for deriving such a conceptualization in a methodical and well-documented way.

Over time, GT has evolved into several variants, most notably Glaserian GT (developed by Glaser et al. [75]), Straussian GT (developed by Strauss et al. [208]), and Constructivist GT (developed by Charmaz et al. [42]). Stol et al. [207] explore the commonalities and differences among the GT variants from a software-engineering perspective. In this chapter, we employ Straussian GT to analyze Canadian food-safety regulations. We select this variant for two primary reasons: (1) our study addresses a well-defined research question (RQ1, as presented in Section 3.1), and (2) as discussed further in Section 3.4, throughout our study, we require the flexibility to engage with the existing literature, which Straussian GT permits.

Straussian GT has the following phases [164, 207, 208]: **Defining Research Questions:** Researchers start with broad, open-ended questions that guide the investigation. **Theoretical Sampling:** In this phase, researchers select data sources based on their ability to contribute to the emerging theory. Sampling continues until theoretical saturation is achieved, meaning no new insights or concepts emerge. **Open Coding:** This phase involves breaking down the data into discrete parts to identify concepts. Labels are assigned to concepts, and emergent categories are recognized. **Constant Comparisons:** Throughout the process, data is continuously compared to refine categories, identify relationships, and ensure no gaps in the theory. **Memoing:** Researchers document their preliminary ideas about the properties and conceptual relationships between categories through memos, which can include notes, diagrams, or sketches. **Axial Coding:** After establishing categories, axial coding links categories and subcategories, enabling a deeper understanding of their interrelationships. **Selective Coding:** Finally, a core category is identified, and the various categories are integrated into a coherent theory that addresses the research question.

All variants of GT, including Straussian GT, are inherently iterative, meaning that one may revisit earlier phases as new insights emerge. For example, in Straussian GT, one might return to open coding or adjust theoretical sampling based on discoveries made during axial or selective coding. This non-linear approach also allows different phases to

occur in parallel, such as memoing throughout the entire process or engaging in axial and selective coding simultaneously as the theory takes shape. This flexibility ensures that the resulting theory is properly grounded in the data and remains responsive to the evolving research context.

3.2.3 Large Language Models (LLMs)

We use LLMs to classify information in food-safety regulations. LLMs are statistical models trained to predict and generate coherent, contextually relevant text. These models are broadly categorized into generative models, like GPT [185], which are optimized for text production, and discriminative models, such as BERT [52], which are optimized for classification and analysis tasks. LLMs are typically pre-trained on large volumes of textual data to learn contextual information, language regularities, and syntactic and semantic relationships. Following pre-training, LLMs usually require *fine-tuning* to achieve optimal performance in downstream tasks [224]. Fine-tuning involves adjusting a (pre-trained) model to tailor it to a specific application or domain.

Generative AI models, including generative LLMs, commonly use *prompts* as input instructions to guide the model, enabling users to specify the desired nature or context of the generated output [94]. Various prompting strategies, such as zero-shot, few-shot, chain-of-thought, and tree-of-thought prompting, exist to enhance output quality [146].

Below, we briefly discuss the LLMs explored in this chapter.

Bidirectional Encoder Representations from Transformers (BERT)[1] employs bidirectional training of transformers – a popular attention model for language modelling [52]. BERT comes in two sizes: *base* and *large*. While BERT large has better accuracy in certain tasks, it is computationally more expensive. BERT can handle capitalization distinctions; the *uncased* version converts text to lowercase before tokenization, while the *cased* version preserves capitalization. Prior research favours the cased model for analyzing requirements and regulatory documents [112, 152]. We therefore adopt the cased model. BERT has several variants that aim to enhance the performance of the original BERT model. In addition to BERT base and BERT large, we experiment with the following BERT variants in this chapter: *ALBERT* [145] and *RoBERTa* [241].

Generative Pre-training Transformer (GPT) is a transformer-based family of models, pre-trained to predict the next token in a sequence [185]. GPT has shown the ability to capture subtle linguistic patterns and perform well in many language tasks [39, 174]. In this chapter, we use GPT-3.5-turbo and GPT-4o for experimentation. These are our best choices as of this writing: The models are both recent and fine-tunable at costs that would allow for the extensive empirical examination necessitated by our experimental procedure.

[1]With the rapid development of generative models, now commonly having billions of parameters, the term "LLM" is evolving and is increasingly used interchangeably with generative language models. Although BERT was not designed for text generation, its architecture, scale, and linguistic capabilities share significant similarities with modern generative models like GPT. Therefore, in this chapter, we include BERT under the term LLM.

We consider two other LLM families – Llama and Mixtral – in addition to BERT and GPT. These are introduced below but are ultimately excluded from our evaluation due to technical limitations, as detailed in Section 3.6.5.

Large Language Model Meta AI (Llama) is a group of open-weight LLMs developed by Meta [216], [124] for different natural language processing tasks. In this chapter, we use Llama-3-8B-Instruct, a model specifically fine-tuned for following instructions.

Mixtral is the first open-weight sparse mixture-of-experts (sMOE) model developed by Mistral AI [133], [121]. By activating only the relevant parameters based on the task, Mixtral optimizes computational efficiency while maintaining high accuracy. In this chapter, we explore Mixtral-8x7B-Instruct-v0.1, a variant specifically fine-tuned for instructional tasks.

3.2.4 Long Short-Term Memory (LSTM)

An alternative model for classification is long short-term memory– a type of recurrent neural network designed to accurately process long sequences of data [113]. Compared to LLMs, LSTM models have fewer parameters and do not require as much computational power to train and run. LSTM is not a part of our solution. Rather, we use LSTM as a baseline to assess whether employing the more computationally expensive LLMs is justified for our intended application. We build our LSTM baseline using Bidirectional Long Short-Term Memory (BiLSTM). BiLSTM uses information from both past and future elements in an input sequence by processing the sequence in both forward and backward directions using two separate LSTM layers [90].

3.3 Related Work

We discuss previous work on (1) extracting metadata from legal texts, (2) applying LLMs to natural-language requirements, and (3) monitoring food safety through IoT. To our knowledge, no prior work exists on automated classification of requirements-related provisions in food-safety regulations.

3.3.1 Metadata Extraction from Legal Texts

Legal documents are typically lengthy and complex, making it difficult for individuals to locate relevant information quickly and accurately. This has led to the development of automated approaches for information extraction from legal texts.

Breaux et al. [34, 36] use semantic parameterization to extract rights and obligations from regulatory texts, and further propose an approach for traceability during requirement extraction [35]. Bhatia et al. [29] address privacy-policy incompleteness by representing data actions as semantic frames. These works, despite being amongst the first to focus on

information extraction from legal texts, do not use Machine Learning (ML) techniques and are limited to privacy policies.

Amaral et al [12] employ a combination of NLP and feature-based learning to extract metadata from privacy policies and assess their compliance with GDPR. In subsequent research [10,13,127], the approach is expanded to incorporate metadata related to GDPR's data processing agreements (DPAs). These prior studies differ from ours in both domain and the utilization of feature-based learning as opposed to LLM-based approaches.

Some previous studies link privacy policies to software code. Fan et al. [61] use traditional ML classifiers to check Android mobile health app compliance with GDPR, while Hamdani et al. [93] combine ML and rules for compliance checking of privacy policies with GDPR. Xie et al. [238] employ Bayesian classifiers and NLP to check whether Amazon Alexa's skills meet their privacy requirements. In contrast, our approach focuses on food-safety regulations. We evaluate models from four families of LLMs – BERT, GPT, Llama, and Mixtral – for the classification of these regulations.

Zeni et al. [239, 240] and Sleimi et al. [202] extract semantic metadata from legal texts using traditional NLP and semantic web techniques. Abualhaija et al. [3, 4] employ LLMs for legal question answering and the identification of regulatory changes, respectively. And, Hassani e al. [109] use LLMs for legal compliance checking. These studies differ from ours in terms of analytical goals, inputs, and the application of NLP and LLM techniques.

3.3.2 LLMs and Natural-language (NL) Requirements

Several studies use LLMs for analyzing NL requirements, with tasks including classifying non-functional requirements [9, 43, 112], classifying and summarizing contractual obligations [128, 191], detecting requirements smells [92], classifying security requirements [222], classifying user feedback [161], classifying requirements dependencies [51], detecting causality [67], classifying and clustering coreferences [228,229], transforming NL requirements into formal specifications [167], predicting issue links [151], classifying issue sentences [160], identifying similar requirement [1], checking completeness [152], generating elicitation-interview scripts [79], named entity recognition in software specification documents [49], detecting variability in NL requirements documents [63], formal requirements engineering [206], requirements quality assurance [64], and checking requirements satisfiability [195]. These studies demonstrate the versatility of LLMs in requirements-analysis tasks. We have benefited from the best practices in the above-cited strands of work; however, our analytical objectives set us apart.

3.3.3 Food-monitoring Systems

Food monitoring is a crucial area of research for improving food safety and enhancing supply-chain efficiency. Bouzembrak et al. [33] have conducted a systematic literature review to examine the potential of IoT for food safety and quality monitoring, as well as food traceability and supply chain. They observe that most existing IoT research aimed

at these objectives focuses on measurement solutions using sensors and communication technologies. These solutions are technology-driven and not directly linked or traceable to regulations. Our GT study highlights, from a regulatory perspective, the importance of all the core measurement types identified in the literature. The close alignment between the measurements implied by food-safety regulations and the existing measurement technologies confirms the validity of our interpretation of systems/software relevance in the context of food safety. Upon snowballing from the above literature review, we found no research that focuses on legal text processing or on establishing traceability between measurements and regulatory requirements.

Brewer et al. [38] introduce a trust framework, and Person et al. [181] deploy food data trusts, both to facilitate secure information sharing across food-supply chains, aiming for regulatory compliance and enhancing transparency. Pearson et al. [182] further apply digital technologies to decarbonize food systems, focusing on supply-chain transparency and sustainability. These studies align closely with our work's emphasis on the potential of IoT for ensuring food safety. Yet, the studies do not address automated processing of food-safety regulations.

The closest work to ours is by Markovic et al. [157], who propose an ontological model for recording provenance in the food-safety domain and demonstrate its utility in the context of automated provenance generation for food-safety compliance checking. They discuss measurements but limit themselves to temperature, whereas our content model (Fig. 3.4) is far richer.

3.4 Characterization of Requirements-related Information in Food-safety Provisions

In this section, we address RQ1 (as posed in Section 3.1) by conducting a GT study, following the Straussian variant of GT as explained in Section 3.2.2. The GT study was conducted collaboratively by the candidate and main supervisor: a PhD student with two years of experience in qualitative and GT analysis, and a senior researcher with 12 years of experience in these areas. The results were subsequently validated by the co-supervisor.

The GT analysis was carried out through in-person, roundtable-style meetings. The candidate and main supervisor collaborated regularly in the second supervisor's office, while joint discussions with the co-supervisor took place in a university conference room. For data analysis and collaborative coding, we used a shared Google spreadsheet, organized with separate tabs corresponding to different parts of the food-safety regulations. This spreadsheet served as the central workspace for recording codes and constant comparisons. We also used Microsoft Visio to create conceptual diagrams and visual representations of our emerging content model.

Although the candidate did not have formal training in food safety, she developed substantial familiarity with the regulatory landscape over the course of this project. Through several months of iterative analysis, including detailed review of over 3,000 regulatory sen-

tences and supporting documents (e.g., SFCR, FSRG), she gained a working understanding of the structure, terminology, and key concerns in food-safety domain.

Study Context and Data Selection. We base our GT study on food-safety regulations in Canada. This decision is made in view of two factors: (1) The majority of our industry partner's market is in Canada, and (2) Canada has one of the most comprehensive regulatory frameworks for food safety, evidenced by the country's top "Quality and Safety" ranking in the Global Food Security Index (GFSI) [72]. While variations are to be expected in other countries and jurisdictions, we anticipate that the concepts identified in our study should generalize to food-safety systems elsewhere.

In our GT study, we analyze the textual content of SFCR and the FSRG regulations introduced in Section 3.2.1. Our data collection follows a theoretical sampling approach, where we carefully choose additional food-safety provisions to analyze based on emerging concepts and patterns, aiming to quickly reach saturation. Quick saturation minimizes unnecessary data collection, saving time and effort, while also improving analysis by allowing researchers to focus on richer data that has the potential to yield deeper insights [207].

To achieve rapid saturation, we prioritize selecting FSRG regulations that we find more likely to refine or complement SFCR. To this end, we choose the regulations for meat and egg products. These products are inherently more complex to prepare, process, maintain, and trade. As a result, they strike a good balance between managing the resource-intensive GT analysis and increasing the likelihood of identifying the most relevant concepts for food-safety systems. For example, a recurring theme in meat regulations is the need for precise and continuous mass measurement. This theme emerged as a salient concept, with meat being a very rich category of food products in which such measurements are taken. A thorough examination of mass measurement for meat products thus reduced the need to repeatedly assess similar content for other food products.

Specifically, a selected subset of the provisions in these regulations was coded by third-parties, as we explain in Section 5.2. The main distinction is that the authors applied a Grounded Theory (GT) approach. In contrast, the third-parties applied hypothesis coding using the author-defined codes as the basis for their work. No conflicts or new codes emerged from the follow-on coding by third-parties.

Analysis Procedure. We reviewed all the sentences in the SFCR, followed by the FSRG regulations on meat and egg. The goal of this review is to identify any sentences that could potentially have some bearing on systems and software requirements. SFCR and the FSRG regulations have cross-references to other documents, e.g., SFCA [82] and Food and Drug Act [80]. We examined the cited references on an as-needed basis to be able to properly interpret the provisions of SCFR and the FSRG regulations under investigation.

While we did not have a preconceived notion of requirements relevance and kept an open mind about the concepts we could encounter, we were guided by two sources of information:

- A literature survey on IoT food-safety monitoring [33]: This survey outlines five main application areas for IoT in food safety. These areas are: *monitoring, sensing, communi-*

Table 3.1: Summary of initial, selected, and coded sentences from the SFCR and FSRG regulations.

Initial Sentences	Selected Sentences	Coded Sentences
12,240	3,135	688

Initial Sentences: Total number of sentences present in SFCR and FSRG regulations.
Selected Sentences: Sentences that are selected for further examination based on relevance to food-safety systems.
Coded Sentences: Sentences that are fully coded in the GT study.

cation, traceability, and data management. Our initial review was particularly sensitive to the incidence of these concepts in the textual content being analyzed.

- Prior research by the RE community on legal requirements: Several code sets exist for labelling and classifying legal provisions, e.g., [12, 19, 29, 139, 202, 213]. While none are specifically directed at food safety, we found the concepts of *data* and *data actions* (e.g., *collection, retention, and transfer*) from privacy requirements [19, 29] helpful when developing our code set. Ultimately, we concluded that, in a food-safety context, there was insufficient justification for including data actions, since virtually no personal data is involved. All actions we came across in our GT data fell under either collection or retention without any clear distinction made between the two.

As discussed in Section 3.2.2, Straussian GT offers the flexibility to incorporate insights from existing literature, in contrast to the more purist versions of GT (e.g., Glasserian), where consulting prior literature is discouraged [75]. In our case, the flexibility to consult the literature was important, as it enabled us to be better informed about the capabilities of existing IoT technologies for food-safety monitoring and to remain sensitive to the nuances of data handling, which have been extensively studied in the RE community.

Table 3.1 provides statistics on the number of initial, selected, and coded sentences from SFCR and the FSRG regulations on meat and egg. These sources combined constitute a total of 12,240 sentences (i.e., the initial sentences in Table 3.1). A section-by-section review of these sources resulted in the exclusion of 9,105 sentences that were not directly related to the requirements of food-safety systems. Notably, the exclusions encompass the SFCR parts that do not pertain to food products. These parts, along with brief descriptions of their content, are listed in Table 3.2.

After excluding orthogonal content, as discussed above, we were left with 3,135 sentences for further analysis (i.e., the selected sentences in Table 3.1). We began analyzing these sentences in the order of their appearance, starting with SCFR, followed by the FSRG regulation on meat, and then the FSRG regulation on egg. As we progressed through these sentences, we carefully monitored for repetitive content to ensure that our coding focused on material most likely to generate new concepts or refine existing ones. This approach

Table 3.2: SFCR parts excluded from our GT analysis on the basis of being unrelated to food-safety systems.

Part Number	Part Title	Content
7	Recognition of Foreign Systems	This part outlines the criteria for recognizing a foreign state's inspection system, focusing on the pertinent administrative processes.
8	Ministerial Exemptions	This part addresses exemptions granted by the Minister for specific regulatory requirements, often for trade-related purposes like alleviating shortages or testing markets.
15	Transitional Provisions	This part deals with the transition to SFCR from previous regulations and the associated amendments.
16	Consequential Amendments, Repeals and Coming into Force	This part lists the regulations repealed when SFCR was enacted, including the Canada Agricultural Products Act, Meat Inspection Regulations, and food requirements from the Consumer Packaging and Labelling Regulations.

allowed us to increase the diversity of the material examined while keeping the coding process manageable in terms of effort. Ultimately, 688 sentences were subjected to systematic coding, as indicated in Table 3.1.

(3) The geographic origin of a food must, subject to the requirements of any other federal or provincial law, be shown

(a) in close proximity to the name and principal place of business of the person by or for whom the food was manufactured, processed or produced; and

(b) in characters of at least the same height as those in which the information referred to in paragraph (a) is shown.

Figure 3.2: A snippet from Part 11 of SFCR.

320 (1) The grade name of prepackaged fresh fruits or vegetables, other than consumer prepackaged fresh fruits or vegetables, must be shown

(a) on any surface of the container, except the bottom; and

(b) in characters of at least the minimum character height that is set out in column 2 of Schedule 6 for the area of a principal display surface that is set out in column 1.

Figure 3.3: A snippet from Part 12 of SFCR.

To illustrate what we mean by repetitive content, consider the snippets in Figs. 3.2 and 3.3. The first snippet is from Labelling (Part 11 of SFCR), while the second is from Grades and Grade Names (Part 12). Both snippets contain specific information that we classify as *Label Data*. In our GT study, we opted not to code the second snippet, as the first had already been coded. This decision saved time and allowed us to focus on more novel content.

While our GT approach follows the principles of the Straussian method, we used Saldaña's practical guidelines [192] for the detailed procedures of first- and second-cycle coding to ensure consistency across multiple coding cycles. Throughout the coding process, we created a series of *memos*, including notes, conceptual diagrams, and slides, to better capture and present reflections and the development of categories. Initially, first-cycle codes were generated through *open coding*, aimed at identifying and categorizing key concepts related to food-safety systems, complemented by *constant comparisons*. These first-cycle codes were then refined and reorganized through second-cycle *axial coding*. Ultimately, the core category identified through *selective coding* reflects a precise, content-driven characterization of whether a given food-safety provision is relevant to the requirements of (software-intensive) food-safety systems. For instance, during the open coding of provisions P_3 and P_4 from the illustrative example in Fig. 3.1, we identified three preliminary concepts: data associated with food items, the collection of data about food items, and the retention of such data. Through constant comparison, the concept of food-associated data was eventually refined into *Label Data*. Furthermore, recognizing that explicit data actions were unnecessary in our domain (as discussed earlier in this section), data collection and retention were merged into a single concept: *Non-Label Data*. During axial coding, the concepts of *Label Data* and *Non-Label Data* were integrated to form the higher-order concept of *Data*.

Coding Results. Figure 3.4 presents our content model for requirements-related provisions based on our GT study of SFCR and the FSRG regulations. This content model serves as our response to RQ1, posed in Section 3.1. The concepts in the model are the targets for automated classification, as we describe Section 3.5. The model is organized into two levels: *level-1 (L1)* concepts are shaded yellow, and *level-2 (L2)* concepts (i.e., sub-concepts of L1) are shaded grey. Concept definitions are provided in the glossary of Table 3.3. Summary statistics for concept instances identified during our GT study are provided in Table 3.4 and discussed further in Section 3.6.5.

We make four important remarks regarding the content model of Fig. 3.4:

1. **Scope of Measurements.** The systems-and-software significance of measurements is in that they can in principle be made through (IoT) *sensing*. Our content model covers only the measurement types observed in our GT study and is therefore not an exhaustive representation of all possible measurements. Sensing applies to various other quantities, such as pressure, luminosity, and motion, among others. While our GT data does not support the inclusion of additional measurements, it does not rule them out either. A broader GT study may find new measurements relevant for food-safety management. Should other measurement types be found pertinent, we do not foresee saturation prob-

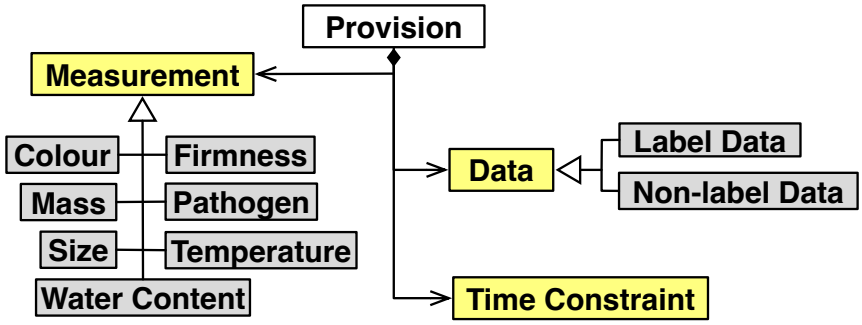


Figure 3.4: Content model for requirements-related provisions in food-safety regulations.

Table 3.3: Glossary for the content model of Fig. 3.4.

Concept	Definition
Data (+sub-concepts)	*Data:* any information used to convey knowledge, provide assurance or perform analysis; *Label Data:* information that a food-product package or container must bear; *Non-label Data:* any food-safety-relevant data other than label data that needs to be collected and/or retained for inclusion in documents such as certificates, reports, guarantees and letters.
Measurement (+sub-concepts)	*Measurement:* Association of numbers with physical quantities; *Colour:* (self-evident); *Firmness:* degree of resistance to deformation; *Mass:* amount of substance by weight or volume; *Pathogen:* a microorganism that causes disease; *Size:* dimension (e.g., length or thickness) or surface area; *Temperature:* (self-evident); *Water Content:* humidity or moisture.
Time Constraint	A temporal restriction, which in our context, is expressed using intervals, deadlines or periodicity.

lems in our content model, noting that the number of fundamental physical quantities that can be measured is limited.

2. **Implicit Concepts.** Although traceability, supply chain and communication are key concepts for food-safety management [33], our GT study did not result in explicit codes for these concepts. Given the technology-neutral nature of SFCR, this finding is not surprising for supply chain and communication, which are technology-laden. The absence of express traceability concepts nevertheless led to further investigation into the underlying reasons. SFCR has a dedicated section on traceability (Part 5). Our follow-up examination of this section confirmed that SFCR establishes traceability requirements primarily through *indirect* means. Specifically, traceability is achieved through the use of information provided on food-product labels (such as lot codes or unique identifiers) and through documentation submitted by distributors, transporters, importers, exporters and retailers. Both types of information are represented in our content model through the concepts of *Label Data* and *Non-label Data*.

3. **Concept Refinement and Consolidation.** The concepts in the model of Fig. 3.4

emerged only after several rounds of refinement. Initially, our set of open codes included the following 18 concepts: *Data Collection*, *Data Retention*, *Data Definition–Food-attached*, *Temperature*, *Dimension Measure*, *Weight/Capacity*, *Water Level*, *Moisture*, *Firmness*, *Bacteria*, *Tests*, *Fat Thickness*, *Colour*, *Volume*, *Deadline/Interval*, *Periodicity*, *Font*, and *Format*. Through iterative analysis, we identified conceptual overlaps, leading to the merging, renaming, or discarding of several initial codes. For example, *Volume* and *Weight/Capacity* were merged into *Mass*, while *Fat Thickness* was discarded, as it was subsumed under the broader concept of *Size*. In total, eight concepts were consolidated into four, and five concepts were discarded. The remaining codes were refined to align more closely with the main regulatory concepts identified: (1) *Dimension Measure* was renamed *Size* to simplify and generalize the concept of physical dimensions for food products, and (2) *Data Definition–Food-attached* was renamed *Label Data* to more accurately reflect its direct application to food products.

4. **Context-driven Codes for Operationalization.** Food-safety regulations focus on *what* must be done, not *how*. For instance, the regulations might require the measurement of *Size* without specifying the method, leaving it open to, say, use calipers for fat thickness versus a thickness gauge for sliced products like cheese and meat. Our GT study reflects this by identifying concepts applicable across various food products. However, different products have unique characteristics, so to generate "lower-level" codes that can help address the "how" for operationalization, one must consider *tuple*-combinations of contextual information about a food product and a high-level code from our content model of Fig. 3.4. For instance, a tuple such as ("citrus fruit", *Size Measurement*) provides the necessary context for determining the appropriate method of measurement. We observe that no significant human effort is required to identify the relevant food product in a specific provision: When a provision targets a particular product, this is clearly indicated by the structure of the regulation. Typically, section titles help analysts easily distinguish between provisions for categories like meat, egg, and fruits. Thus, lower-level codes can be inferred from the *Cartesian product* of food products and the high-level codes identified in our GT study. Explicitly enumerating the Cartesian product of all potential combinations (food products and high-level codes) would not provide additional insight to our study, as the regulations themselves avoid irrelevant combinations, e.g., ("milk" and *Firmness Measurement*). Ruling out what is illogical or unhelpful (from a regulatory standpoint) is easily achievable once a given snippet of regulatory text for a specific food product has been processed. For instance, after automatically analyzing and labelling the section concerned with dairy products, the user can pose a query such as "Are there any statements in this content that have been annotated by the concept of *firmness*?

3.5 Automated Classification of Food-safety Provisions

In this section, we present a pipeline for automatic classification of regulatory food-safety provisions based on the concepts identified in our GT study of Section 3.4. Figure 3.5

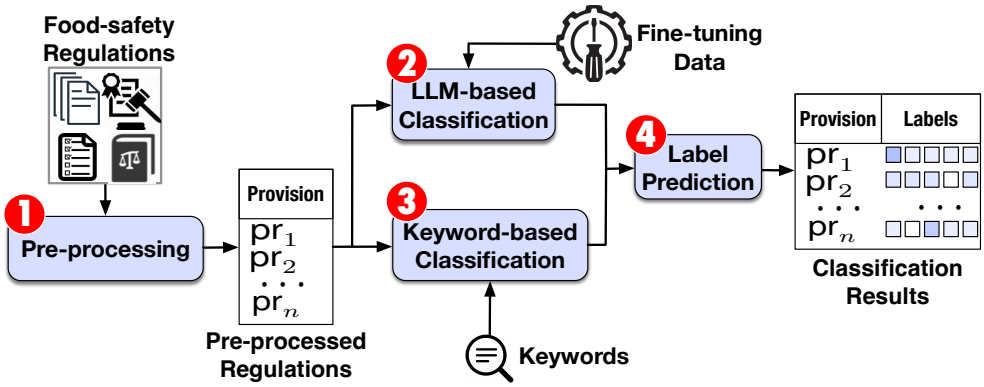


Figure 3.5: Overview of automated classification pipeline.

presents an overview of the pipeline, which takes as input the textual content of the regulations being examined and produces as output labels for each provision in the input.

The pipeline treats each sentence as one provision. Stated otherwise, our *unit of analysis* is a sentence. This decision aligns with previous research on automated processing of legal texts [12], which found that sub-sentence localization offers limited additional value, while larger units like paragraphs and sections may lack precision. Although Amaral et al. [12] focus on data-protection regulations, we observe that food-safety regulations share key characteristics with other legal frameworks, including data protection. The similarities notably include formal legal language, defined roles, distinctions between permissions and obligations, and accountability criteria. In food safety, actors such as food producers, manufacturers, and regulatory bodies (e.g., the Canadian Food Inspection Agency) ensure product safety, proper labelling, and hygiene. Likewise, data protection frameworks assign roles to data controllers and processors, overseen by data protection authorities, as in the General Data Protection Regulation (GDPR). Both regulatory frameworks differentiate between permissible actions, like using approved food additives or processing personal data lawfully, and obligatory actions, such as maintaining hygiene or protecting personal data. Accountability mechanisms, such as audits and inspections, are central to both. These commonalities support using sentences as natural units of analysis for classifying food-safety regulations.

Our pipeline has four steps, as we describe next.

3.5.1 Pre-processing

In this step, we split the input into sentences. The term "sentence" does not necessarily refer to a grammatical sentence, but rather to a text segment that has been identified as a sentence by the sentence splitter module in the NLP pipeline. Legal documents frequently use lists. To ensure preservation of context during sentence splitting, we follow the approach of Sleimi et al. [202] and add the header in each list, known as a list-item prefix, to each itemized/enumerated item.

3.5.2 LLM-based Classification

Using an LLM customized using labelled data from our GT study (Section 3.4), we classify provisions with occurrences of concepts from the model of Fig. 3.4. In our evaluation (Section 3.6), we consider four families of models – BERT, GPT, Llama, and Mixtral – as alternatives. For the generative LLMs in our evaluation, the pre-processed regulations from Step 1 are embedded into prompts, as explained in Section 3.6.4.

3.5.3 Keyword-based Classification

Our GT study did not yield a sufficiently large number of examples for *Colour*, *Firmness*, *Pathogen* and *Water Content*. As previous studies on automated classification of legal texts point out [12, 202, 213], some concepts may turn out to be too scarce for learning-based techniques, thereby warranting the use of keywords for classification. In the third step of our approach, we perform a keyword lookup for the above four concepts, for which we found too few ($\leq$10) instances in our GT study. The keywords associated with each concept were derived during our GT study and are listed in our online material [101]. If a provision P contains one or more of the keywords associated with a concept M, we label P with M. For instance, if a provision contains the keywords "E. coli" or "Salmonella Enterica", we label it with *Pathogen*. We note that this keyword-based strategy is reserved exclusively for sparse concepts due to the substantial effort needed to develop comprehensive keyword lists as well as the risk of overfitting.

3.5.4 Label Prediction

This step combines the labels computed by the LLM-based classifiers (Step 2) and those computed by the keyword-based classifiers (Step 3) to produce the final label recommendations for each provision.

3.6 Empirical Evaluation

In this section, we instantiate the classification pipeline of Fig. 3.5 using various LLMs and customization strategies, and then evaluate the accuracy of each resulting configuration.

In our empirical evaluation, we compare models from the BERT and GPT families of LLMs alongside simpler traditional classifiers. While these model families differ substantially in their architectures, capabilities, and typical applications—BERT-based models being discriminative and designed primarily for classification tasks, GPT-based models being generative and optimized for natural language generation tasks, and traditional methods being more straightforward yet computationally efficient—the comparison remains fair and meaningful within our specific context. All models were tested under identical experimental conditions, using the same dataset, labelling schema, preprocessing methods, and evaluation metrics, ensuring that any observed performance differences directly reflect

the strengths and limitations of the respective approaches for this particular classification problem. Including traditional methods provides essential baselines, illustrating the practical trade-offs between model complexity and performance, as supported by previous research [71].

3.6.1 Implementation

Our classification pipeline is implemented in Python. The sentence splitter is implemented using SpaCy 3.4.4. The BERT models we experiment with are: *albert-base-v1*, *bert-large-cased*, *bert-base-cased*, and *roberta-base*, all from Hugging Face [116] and operated in PyTorch 1.13.1. To optimize the fine-tuning hyperparameters of BERT, we use Optuna 3.1.0 [7]. The GPT models we experiment with are *GPT-3.5-turbo* and *GPT-4o*, the Llama model is *Llama-3-8B-Instruct*, and the Mixtral model is *Mixtral-8x7B-Instruct-v0.1*. For fine-tuning GPT, we use the OpenAI API through the OpenAI Python package. For fine-tuning Llama and Mixtral, we use Transformers 4.40.1, BitsAndBytes 0.42.0 for 4-bit model quantization, PEFT 0.12.0 for LoRA-based fine-tuning, and TRL 0.9.6 for supervised fine-tuning.

We further implement two baseline approaches for comparison: one utilizing the (Bi)LSTM architecture [90] and the other using automatic keyword extraction. The BiLSTM baseline is implemented in PyTorch 1.13.1, with the required NLP pipeline and word embeddings built using Keras 2.11.0 and GloVe [183]. We use the same approach for hyperparameter optimization of the BiLSTM baseline as we did for BERT. Our keyword-based baseline employs the Python-based implementation of Arora et al. [23]'s keyword extraction approach, available as part of the WikiDoMiner tool [59]. This baseline trains a Random Forest (RF) classifier to learn how to associate certain keyword combinations with a specific label. Once trained, the classifier can be used to predict the label of new sentences. This baseline further uses scikit-learn 1.2.0 to implement both its Count Vectorizer and RF classifier. We note that this baseline is not comparable to Step 3 of our classification pipeline (Section 3.5), which requires manually defined keywords and is reserved for sparse concepts only. In addition, we use scikit-learn 1.2.0 for dataset splitting and metrics calculation throughout our evaluation. As a technical remark, it is worth mentioning that for BERT and our two baselines, we have structured our requirements-relevance classification problem – a multi-label classification task – as a series of binary classification tasks, with each binary classifier predicting the presence or absence of one specific concept within provisions.

3.6.2 Data Collection Procedure

Our evaluation dataset consists of two distinct components: a fine-tuning/training set, denoted F, and a test set, denoted T. Set F comes from our GT study of Section 3.4. For further information on our GT study that led to the development of F, we have made available our initial labelling document, which includes guidelines, colour-coded sheets, raised questions, notes, and relevant links, as part of our Appendix A and online material

[101]. Below, we describe the procedure for building T. As explained in Section 3, we excluded from our qualitative study all FSRG regulations except those on meat and egg products. To construct T, we include the following FSRG regulations: dairy products, fish, fruits and vegetables, honey, manufactured products, and maple. These are all the FSRG regulations, excluding those on meat and egg, which were part of our GT study. In addition, we include selected food-safety regulations from the US FDA. The motivation to augment our test set with FDA regulations is to assess the generalizability of our pipeline over food-safety regulations in a jurisdiction different from that covered by the documents examined in our GT study. This augmentation is also natural and useful, considering that, like Canada, the US ranks highly in "Quality and Safety" according to GFSI [72], making it a valuable source for evaluating the thoroughness of our classification experiments.

For the FDA portion, we consider three sources under the food category: *packaging* [218], *labeling* [221] and *guidance for industry* [219, 220]. To create T, we randomly select 400 provisions (sentences). Of these, 350 are from the FSRG regulations excluded from our GT study, and the remaining 50 from the FDA sources cited above. To create an unbiased ground truth for testing, we collaborated with two third-party annotators. Both are graduate students in Computer Science. They possess excellent English language skills, have 2+ years of prior industry experience, and have been exposed to requirements engineering and qualitative coding in their graduate coursework. The annotators were paid for their work.

We conducted a two-hour training session for the annotators before they started their work. The training covered food-monitoring systems, food-safety management and the content model of Fig. 3.4. To label provisions according to this content model, the annotators were instructed to apply hypothesis coding – a qualitative coding approach in which pre-existing codes guide the analysis [192]. In this case, the pre-existing codes are those from the content model of Fig. 3.4. The annotators received a protocol document to use as a reference, explaining concepts of interest with examples; this protocol document is available in our online material [101]. While the annotators were given the flexibility to extend our codes if needed to incorporate new insights, no new codes emerged from their work.

We divided the 400 provisions in T equally between the two annotators, including a 10% overlap to assess annotation reliability. That is, each annotator examined 210 provisions (190 distinct and 20 shared). We use Cohen's kappa (κ) for measuring inter-rater agreement. For a concept M, a provision P counts as an agreement if both annotators make the same decision as to whether M applies to P. Divergent decisions count as disagreements. The disagreements were resolved through consensus between the two annotators, following discussion.

3.6.3 Evaluation Metrics

We evaluate accuracy using *Precision* and *Recall*, with their standard definitions. For every classification label M, the definitions of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) are as follows: A *TP* is a prediction of `true` for a provision that has M in its label set (as per the ground truth); a *FP* is a prediction

of true for a provision that does not have M in its label set; a TN is a prediction of false for a provision that does not have M in its label set; and a FN is a prediction of false for a provision that has M in its label set.

Precision and Recall are often reported alongside the F *measure* (or F *score*) – the harmonic mean of Precision and Recall – which provides a unified metric for classification accuracy. For clarity, we discuss accuracy directly in terms of Precision and Recall. F-measure results are available in our online material [101].

We use the non-parametric Wilcoxon Rank-Sum Test [41], also known as Mann-Whitney U Test, to compare the accuracy of different algorithms, specifically different classification pipelines in our context. Through this test, we assess whether there is a significant difference in the distribution of accuracy metrics between two algorithms, without assuming any particular underlying data distribution. We note that, in our evaluation, we always compare alternative algorithms by measuring their accuracy on a single test set. The reason we obtain a distribution rather than a single-point accuracy value is the inherent random variation caused by the non-deterministic nature of training, fine-tuning, and (potentially) inference in deep learning, which typically leads to a range of accuracy values. For example, even when BERT or GPT is fine-tuned multiple times under the same conditions, randomness in the fine-tuning process can result in different model parameters. This can in turn cause the models to yield different accuracy results when tested on the same dataset. In generative LLMs like GPT, this effect is intensified by hyperparameters such as temperature, which introduce randomness during output generation.

The null hypothesis (H_0) for our statistical tests states that there is no difference in the distribution of the accuracy metric of interest – either *Precision* or *Recall* – between the two algorithms being compared. The goal of these tests is to assess whether the observed differences are statistically significant or merely due to random variation [21].

To assess the practical significance of the difference between two algorithms, we use Vargha-Delaney's non-parametric effect-size statistic, denoted as $\hat{A}_{12}$ [223]. Given a pair of algorithms, the value of $\hat{A}_{12}$ represents the probability that a randomly selected observation from one algorithm will outperform a randomly selected observation from the other. We categorize effect sizes into four levels, based on the widely used threshold-based guidelines from Vargha et al. [223]: *L (Large)*: A strong difference between the algorithms, indicating a clear performance advantage; *M (Medium)*: A moderate difference, suggesting some advantage, though less pronounced; *S (Small)*: A small difference, indicating limited practical difference; and *N (Negligible)*: Almost no practical difference between the algorithms.

3.6.4 Analysis Procedure

EXPI. This experiment answers RQ2 as posed in Section 3.1. The experiment consists of two stages: (1) customizing an LLM using the F portion of the dataset, as defined in Section 3.6.2 and (2) applying the (customized) LLM to the T portion, as defined in the same section. The technical details of these two stages differ depending on the chosen LLM

and whether fine-tuning or few-shot learning is used for customization. We describe the stages separately for BERT, GPT, Llama and Mixtral.

Given the non-determinism in training deep learning models, caused by factors such as initialization and regularization, it is important to consider random variation during both the initial training phase and any subsequent fine-tuning [44, 184]. In generative LLMs like GPT, beyond the stochastic elements of training, inference also involves sampling from a probability distribution. As mentioned in Section 3.6.3, hyperparameters such as temperature, which determine how deterministic or creative the output of an LLM is, introduce additional variability. To account for random variation, we conduct 20 independent experiments for each concept M in our content model (Fig. 3.4) and present the results as boxplots[2]. We opted for 20 independent runs rather than the conventional 10 to enhance the statistical reliability of our results. Increasing the number of runs helps mitigate the effects of variability leading to more robust and generalizable performance estimates [188].

BERT. We experiment with four BERT variants: BERT base, BERT large, ALBERT and RoBERTa. To fine-tune each BERT model effectively, we optimize three hyperparameters: learning rate, number of epochs, and choice of optimizer. The ranges explored for learning rate and the number of epochs are $[2e{-}5, 2e{-}4]$ and $[10, 35]$, respectively, in accordance with best practices [52]. For the optimizer, we consider two alternatives: SGD [32] and AdamW [148]. As for the batch size and maximum sequence size hyperparameters, we respectively use 16 and 512 – the default values suggested by Devlin et al. [52]. The optimal hyperparameters for each concept of our model are provided online [102]. With the BERT hyperparameters optimized and the most accurate fine-tuned model built for each concept in the first stage, the second stage applies the resulting BERT models to the (third-party-annotated) test set, T.

GPT. We experiment with two variants of GPT: GPT-3.5-turbo and GPT-4o. To adapt these models to our task-specific data, we consider fine-tuning. A natural question that may arise here though is whether fine-tuning powerful LLMs such as GPT is worthwhile, considering that these models already have extensive pre-trained knowledge and generalization capabilities. To be able to address this question, in our evaluation of GPT-4o, we further explore *few-shot learning* as an alternative to fine-tuning.

Fine-tuning is done over three epochs, with the number selected automatically by GPT based on our dataset size. The fine-tuning data, F, is fed to GPT in a conversational chat format, where examples are structured as messages containing roles for the system, user, and assistant [175]. The system role provides instructions that the model should follow; the user role presents input that the model should respond to; and, the assistant role represents the model's response to the user's input. In this setup, the user role presents input paragraphs, and the assistant role outputs the labelled sentences. After fine-tuning, the fine-tuned models are evaluated against T. In our context, the system role instructs the model to label the sentences based on the identified concepts. The user role and the assistant role present input paragraphs and the labelled sentences, respectively. Our

[2]As we discuss in Section 3.6.5, challenges with Llama and Mixtral in getting them to follow instructions prevented systematic experimentation with these two LLM families. Therefore, we provide detailed results only for BERT and GPT.

prompt for fine-tuning is shown in Fig. 3.6.

We set GPT's temperature hyperparameter to a low value, 0.2, following OpenAI's guidelines for situations where one wants to minimize variation in responses and thus make GPT outputs more deterministic [171]. While increased determinism is generally advantageous for classification tasks, absolute determinism is often not ideal, and in fact not technically feasible in GPT either, as per OpenAI's documentation [171]. To this end, we further note that our classification task is text classification. Textual descriptions, especially laws and regulations, often contain ambiguities and context-dependent nuances. Attempting to eliminate non-determinism could therefore prevent the model from accounting for different plausible interpretations.

```
1  We are processing a regulatory document on food safety.
2  Your task is to extract sentences from the following food-safety
3  paragraph and label them based on the concepts they contain.
4
5  The paragraph is enclosed within % delimiters. If a concept is present
6  in a given sentence, label the sentence accordingly.
7
8  Use only the following labels: {list of labels from glossary table}.
9  The concepts are defined as follows: {definitions from glossary table}.
10 Paragraph:{%the target food safety paragraph%}
11
12 Answer: {Correct answer based on the fine-tuning dataset (F)}
```

Figure 3.6: Prompt used for fine-tuning. Once the model is fine-tuned, to have it predict labels for a given paragraph, line 12 needs to change to "Answer." (in the imperative form).

For few-shot learning, GPT-4o is provided with a small set of examples to help it understand the classification task. Specifically, we include in our prompt five representative examples taken from the fine-tuning data, F. Our few-shot prompt (excluding the five examples) is shown in Fig. 3.7. We note that explicit output formatting instructions are not included in either the fine-tuning or few-shot prompts. This is because the examples provided in both contexts capture both the classification task *and* the desired output format, allowing us to keep output formatting implicit.

Llama and Mixtral. We experiment with one variant of Llama, specifically Llama-3-8B-Instruct [124], and one variant of Mixtral, specifically Mixtral-8x7B-Instruct-v0.1 [121]. These variants were chosen because they are optimized for instruction-following, and they were the largest models that our institutional computational resources could accommodate. We fine-tune these models over 25 epochs using Parameter-Efficient Fine-Tuning (PEFT) with Low-Rank Adaptation (LoRA). The hyperparameters `r` and `lora_alpha` are set to their default values, 16 and 32, respectively, as provided by Hugging Face's PEFT framework [125]. In both models, the temperature hyperparameter is set to 0.2 (the same value used for the GPT models). The fine-tuning set, F, is fed to the models in a conversational

```
We are processing a regulatory document on food safety.
Your task is to extract sentences from the following food-safety
paragraph and label them based on the concepts they contain.

The paragraph is enclosed within % delimiters. If a concept is present
in a given sentence, label the sentence accordingly.

Use only the following labels: {list of labels from glossary table}.
The concepts are defined as follows: {definitions from glossary table}.

Below are a few examples to illustrate the task:
Example 1: {labeled example 1}
Example 2: {labeled example 2}
Example 3: {labeled example 3}
Example 4: {labeled example 4}
Example 5: {labeled example 5}

Now, apply the same logic to the paragraph below.
Paragraph:{%the target food safety paragraph%}

Answer.
```

Figure 3.7: Prompt used for few-shot learning.

chat format, similar to GPT fine-tuning (Fig. 3.6), but adapted to the specific prompt structures of these models [121, 124]

For Llama, the system, user, and assistant roles are structured similarly to GPT, but with model-specific tokens to distinguish them in the input prompt. In contrast, Mixtral uses a simpler format that does not explicitly separate system and user roles. Instead, the prompt consists of an instruction followed by the model's response. Detailed prompts for both Llama and Mixtral can be found in our online material [103].

EXPII. This experiment answers RQ3, as posed in Section 3.1. We compare our LLM-based classification models against two non-Transformer (and simpler) models, namely *BiLSTM* and *automatic keyword extraction*, as we outline below.

The input to the BiLSTM baseline is a sequence obtained by tokenizing the food-safety provisions to classify. The tokens are represented using GloVe word embeddings [183]. We optimize the hyperparameters of the BiLSTM model and train the model (on F) in a manner similar to BERT, as explained in EXPI. The implementation of our BiLSTM baseline, the set of hyperparameters for this baseline, and the optimal values of the hyperparameters obtained for each concept in our content model are provided online [102].

Our keyword-based baseline is inspired by the well-known bag-of-words classifier [60, 134]. Instead of using tokens as features, however, this baseline uses keyphrases to represent

the content. The baseline first uses the approach of Arora et al. [23] to extract domain-specific keyphrases from the F portion of our dataset. It then builds a Random Forest (RF) classifier (over F) using the extracted keyphrases as features.

Like the models in EXPI, our baselines are affected by random variation. In the case of BiLSTM, the source of randomness is the same as that for BERT models. For Keyword Search, the randomness is caused by the "random state" variable in the RF algorithm [197]. To account for randomness, we employ the same methodology as in EXPI, running each baseline 20 times on the test set T and reporting boxplot statistics.

3.6.5 Results

Dataset

We recall from Section 3.6.2 that our dataset is comprised of a fine-tuning/training set, F, and a test set, T. Set F resulted from our GT study (Section 3.4) and set T was developed by third-parties. We observed only one discrepancy in the level-2 labels that the annotators provided for the overlapping part of their tasks. One annotator identified *Label Data* in one of the provisions whereas the other annotator did not. This resulted in a Cohen's κ of $\approx$0.90 (almost perfect agreement) for *Label Data*. All other κ values are 1 when at least one concept instance was present. Table 3.4 provides summary statistics for the incidence of different concepts in F and T. For T, we further show the breakdown between labels over the FSRG and FDA regulations.

Of the 688 provisions in F, 369 ($\approx$54%) have some label attached to them; and, of the 400 provisions in T, 224 (56%) have some label attached. The *Overall* label in Table 3.4 represents a higher-level concept implied by the presence of more specific concepts in the content model of Fig. 3.4. More precisely, a given provision receives the *Overall* label if and only if it has been marked with at least one concept from the content model. Essentially, *Overall* distinguishes between provisions that have implications for software (i.e., those annotated with some concept from the model of Fig. 3.4) and those that are not relevant to software. The ability to make this distinction directly supports Use Case 1, as discussed in Section 3.1 (Target Users and Practical Applications). For classification, we treat *Overall* as an independent label. Our pipeline classifies regulatory provisions based first on their *Overall* relevance to food-safety monitoring systems (relevant or non-relevant) and further categorizes relevant provisions according to specific regulatory concepts from our GT-derived content model (Fig. 3.4).

Four concepts, *Colour*, *Firmness*, *Pathogen* and *Water Content* are scarce in our GT study. As noted in Section 3.5.3, we consider using keywords identified in our GT study to classify these four concepts.

Table 3.4: Distribution of labels in our dataset.

Label (Level)	**GT Study (F)**	**Third-party Annotations (T)**	
		Canada (FSRG Reg.)	**USA (FDA)**
Colour (L2)	10	10	0
Data (L1)	178	69	11
Firmness (L2)	5	2	0
Label Data (L2)	121	53	10
Mass (L2)	94	65	32
Measurement (L1)	165	109	33
Non-label Data (L2)	57	18	1
Pathogen (L2)	10	4	0
Size (L2)	32	36	0
Temperature (L2)	31	13	1
Time Constraint (L1)	44	9	8
Water Content (L2)	3	4	0
Overall	369	184	40

Table 3.5: Accuracy of BERT variants for the *Overall* concept

Language Model	Precision		Recall	
	Avg	**SD**	**Avg**	**SD**
BERT base	**0.87**	0.02	0.86	0.03
BERT large	0.86	0.03	0.85	0.04
ALBERT	0.85	0.04	0.84	0.04
RoBERTa	0.84	0.01	**0.87**	0.08

RQ2: How accurately can LLMs classify requirements-related provisions in food-safety regulations?

Identifying the most accurate BERT variant. Table 3.5 presents the average (Avg) and standard deviation (SD) for accuracy metrics, calculated over 20 runs of the four BERT variants under consideration, on the test set, T. Hyperparameter optimization and fine-tuning for all variants followed the procedure in Section 3.6.4. Among the variants, with one exception, BERT base yields the best accuracy metrics for the *Overall* concept. The exception is RoBERTa's *Recall*, which is 1% better than that of BERT base. However, this gain in *Recall* is small and without statistical significance. In general, if one considers the metrics of Table 3.5 for the *Overall* concept only, there is little difference to note between the alternatives in Table 3.5. A closer examination of the behaviour of the variants on individual labels nonetheless reveals that BERT base has the least amount of random variation. We select BERT base as the best-performing variant for further comparison with generative LLMs later in this section. We refer the reader to our online material [102] for detailed results on the other three BERT variants.

Challenges with Llama and Mixtral. We spent significant time and computational resources on fine-tuning and testing the Llama and Mixtral variants considered (Llama-3-8B-Instruct [124] and Mixtral-8x7B-Instruct-v0.1 [121]). However, we were unable to achieve satisfactory results. These models frequently introduced unnecessary justifications, repeated their responses, and paraphrased or generated content not found in the input. These issues made systematic evaluation impossible because the outputs were unpredictable. We believe these challenges stem from the complexity of our classification task, which requires: (1) reading a text passage and extracting sentences based on knowledge learned during fine-tuning on the training set, (2) understanding our content model's customized definitions, and (3) accurately assigning correct hierarchical labels.

As a result, we exclude Llama and Mixtral from RQ2 and do not report accuracy statistics for them. Our analysis of generative LLMs therefore focuses on the GPT family and compares it with BERT base, the best-performing model within the (non-generative) BERT family.

Accuracy statistics for BERT base, GPT-3.5 and GPT-4. Figures 3.8 and 3.9 present the average accuracy metrics from 20 runs for the fine-tuned BERT base, fine-tuned GPT-3.5-turbo, fine-tuned GPT-4o, and GPT-4o with few-shot learning. These results were obtained following the evaluation procedure explained in EXPI (Section 3.6.4).

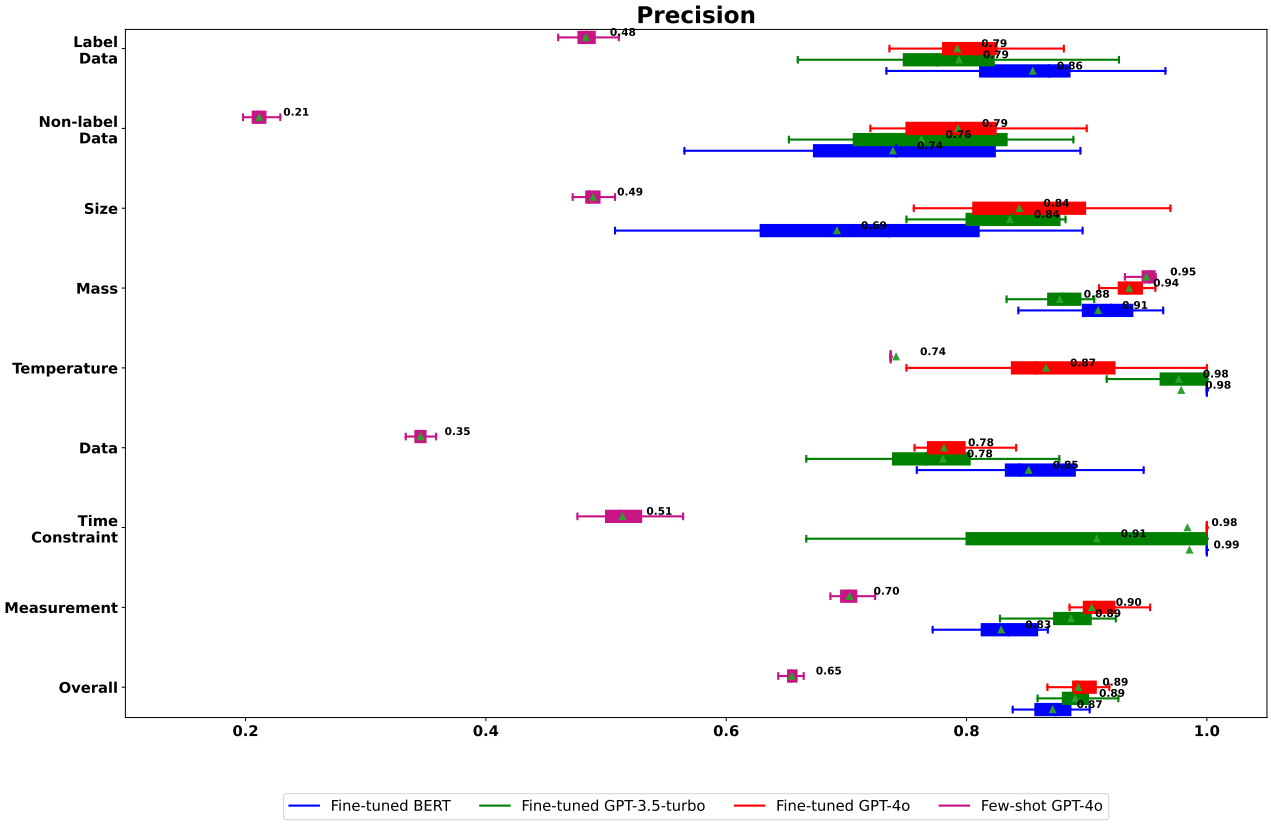


Figure 3.8: *Precision* for fine-tuned BERT base, fine-tuned GPT-3.5-turbo, fine-tuned GPT-4o, and GPT-4o with few-shot learning (the x-axis represents the *Precision* values).

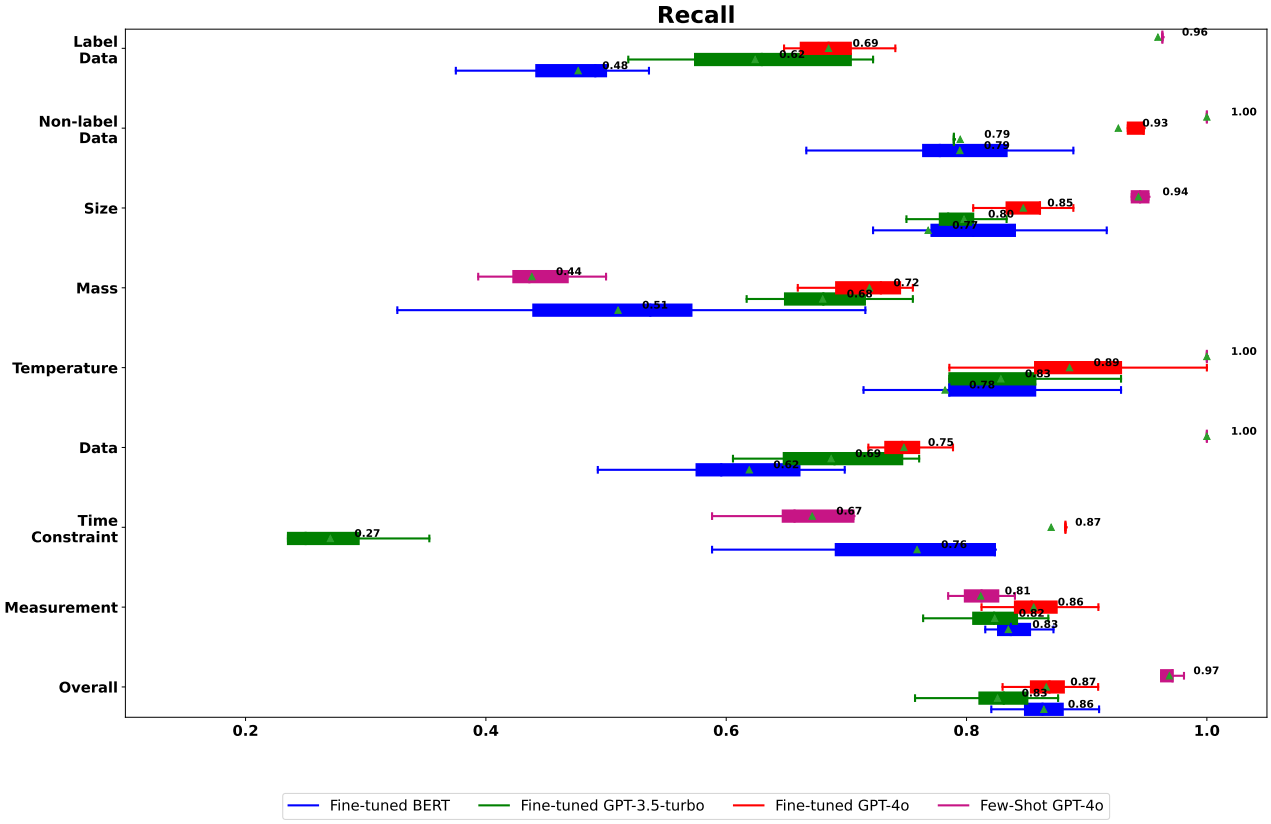


Figure 3.9: *Recall* for fine-tuned BERT base, fine-tuned GPT-3.5-turbo, fine-tuned GPT-4o, and GPT-4o with few-shot learning (the x-axis represents the *Recall* values).

Table 3.6: Statistical comparisons for RQ1 and RQ2.

			Overall		Measurement (L1)		Data (L1)		Time Constraint (L1)	
			p-value	Â12	p-value	Â12	p-value	Â12	p-value	Â12
RQ2	Fine-tuned BERT vs. Fine-tuned GPT-3.5	Precision	<0.05	0.23 (L)	<0.05	0.07 (L)	<0.05	0.84 (L)	0.09	0.66 (M)
		Recall	<0.05	0.81 (L)	0.1	0.65 (M)	<0.05	0.2 (L)	<0.05	1 (L)
	Fine-tuned BERT vs. Fine-tuned GPT-4	Precision	<0.05	0.19 (L)	<0.05	0.06 (L)	<0.05	0.86 (L)	0.81	0.52 (N)
		Recall	0.63	0.45	<0.05	0.3 (M)	<0.05	0.05 (L)	<0.05	0.06 (L)
	Fine-tuned GPT-4 vs. Fine-tuned GPT-3.5	Precision	0.27	0.6 (S)	<0.05	0.72 (L)	0.69	0.54 (N)	0.09	0.65 (M)
		Recall	<0.05	0.84 (L)	<0.05	0.81 (L)	<0.05	0.83 (L)	<0.05	1 (L)
	Fine-tuned GPT-4 vs. Few-shot GPT-4	Precision	<0.05	1 (L)	<0.05	1 (L)	<0.05	1 (L)	<0.05	1 (L)
		Recall	<0.05	0 (L)	<0.05	0.94 (L)	<0.05	0 (L)	<0.05	1 (L)
RQ3	Fine-tuned BERT vs. BiLSTM	Precision	0.12	0.64 (M)	0.055	0.32 (M)	<0.05	0.99 (L)	<0.05	0.72 (L)
		Recall	<0.05	0.99 (L)	<0.05	1 (L)	<0.05	0.99 (L)	<0.05	0.99 (L)
	Fine-tuned BERT vs. Keyword Search	Precision	<0.05	1 (L)	0.65	0.54 (N)	<0.05	1 (L)	<0.05	1 (L)
		Recall	<0.05	1 (L)	<0.05	1 (L)	<0.05	1 (L)	<0.05	1 (L)
	Fine-tuned GPT-4 vs. BiLSTM	Precision	<0.05	0.79 (L)	<0.05	0.85 (L)	<0.05	0.96 (L)	<0.05	0.71 (L)
		Recall	<0.05	0.99 (L)	<0.05	1 (L)	<0.05	1 (L)	<0.05	1 (L)
	Fine-tuned GPT-4 vs. Keyword Search	Precision	<0.05	1 (L)	<0.05	0.95 (L)	<0.05	1 (L)	<0.05	1 (L)
		Recall	<0.05	1 (L)	<0.05	1 (L)	<0.05	1 (L)	<0.05	1 (L)

Effect size: Large (L), Medium (M), Small (S), Negligible (N)

In Table 3.6, we statistically compare the accuracy of the most practically relevant pairs selected from these four models for both the *Overall* concept and the level-1 (L1) concepts, as shown in Fig. 3.4. We do so using the Wilcoxon Rank-Sum Test and the Vargha-Delaney effect size, $\hat{A}_{12}$, as described in Section 3.6.3. The p-values in Table 3.6 assess whether the difference between two models is statistically significant. A p-value less than 0.05 (i.e., $p < 0.05$) suggests that the observed accuracy difference between the models is statistically significant, meaning it is unlikely to have occurred by chance. To interpret the $\hat{A}_{12}$ values, note that in a comparison between $Model_A$ and $Model_B$, i.e., "$Model_A$ vs. $Model_B$", the value of $\hat{A}_{12}$ is directional: $\hat{A}_{12} > 0.5$ favours $Model_A$, while $\hat{A}_{12} < 0.5$ favours $Model_B$.

Comparing the fine-tuned models. Starting with the three fine-tuned models, we observe from the results for the *Overall* concept that, in terms of *Precision*, fine-tuned GPT-3.5-turbo and fine-tuned GPT-4o significantly outperform fine-tuned BERT base, with averages of ≈89% each, compared to fine-tuned BERT's ≈87%. For *Recall*, fine-tuned BERT base and fine-tuned GPT-4o significantly outperform fine-tuned GPT-3.5-turbo, with averages of ≈87% and 86%, respectively, compared to fine-tuned GPT-3.5-turbo's ≈83%.

Based on the above, fine-tuned GPT-4o is a better alternative to fine-tuned GPT-3.5-turbo for the *Overall* concept in terms of both *Precision* and *Recall*. This advantage extends to the more specific L1 concepts of *Measurement*, *Data*, and *Time Constraint*. As shown in Figs. 3.8 and 3.9, and Table 3.6, fine-tuned GPT-4o significantly outperforms

fine-tuned GPT-3.5-turbo in *Recall* across all L1 concepts. For *Precision*, fine-tuned GPT-4o has a statistically significant advantage over fine-tuned GPT-3.5-turbo in *Measurement*. In the cases of *Time Constraint* and *Data*, fine-tuned GPT-4o's average *Precision* is either equal to or slightly better than GPT-3.5-turbo's, though these differences are not statistically significant. Similar trends are observed for the L2 concepts, with statistical results provided in our online material for brevity [102]. Taken together, our findings indicate that when models are fine-tuned, GPT-4o offers an advantage over GPT-3.5-turbo for our classification task.

With fine-tuned GPT-4o shown to outperform fine-tuned GPT-3.5-turbo, the next question is how it compares to fine-tuned BERT base. As stated earlier, for the *Overall* concept, fine-tuned GPT-4o significantly outperforms fine-tuned BERT base in *Precision*, though neither model shows a statistically significant advantage in *Recall*. At the L1 concept level, fine-tuned GPT-4o significantly outperforms fine-tuned BERT base in *Precision* and *Recall* for *Measurement*, *Recall* for *Data*, and *Recall* for *Time Constraint*. Fine-tuned BERT base, on the other hand, has significantly higher *Precision* for *Data*. No other significant differences were observed.

Since fine-tuned BERT base never outperforms fine-tuned GPT-4o in *Recall*; its higher *Precision* for *Data* is offset by lower *Recall*; and similar trends are observed for L2 concepts (with statistical results provided in our online material for brevity [102]), fine-tuned GPT-4o presents the better alternative in comparison with fine-tuned BERT base as well.

In summary, considering that fine-tuned GPT-4o presents a better alternative than both fine-tuned GPT-3.5-turbo and fine-tuned BERT base without any major trade-offs, we conclude that GPT-4o is the most suitable model for our classification task in the fine-tuning scenario.

Comparing GPT-4o fine-tuning and few-shot learning. As shown in Figs. 3.8 and 3.9, for the *Overall* concept, GPT-4o with few-shot learning achieves an average *Precision* of ≈65%, about 24% lower than fine-tuned GPT-4o. However, GPT-4o with few-shot learning achieves a notably high *Recall* of ≈97%, outperforming fine-tuned GPT-4o by about 10%. This trend largely extends to the L1 concepts: fine-tuned GPT-4o shows higher *Precision* across all three L1 concepts, while GPT-4o with few-shot learning achieves better *Recall* for two (*Data* and *Time Constraint*), but lags by 5% over *Measurement* (≈81% vs. ≈86%). All differences are statistically significant with large effect sizes, as shown in Table 3.6.

The above results highlight an important trade-off: For our classification task, while fine-tuning improves *Precision*, it generally has a negative impact on *Recall*. Interestingly, all the examples used in our few-shot learning experiment are drawn from our fine-tuning/training set, meaning that our fine-tuned models are exposed to all the few-shot examples (along with many others). The phenomenon observed here aligns with findings in the literature on LLMs, which suggest that providing a large number of specific examples during fine-tuning may lead to overfitting, causing the fine-tuned model to pick up on spurious correlations and also potentially reducing its generalization abilities [163].

From a practical standpoint, fine-tuning a powerful model like GPT-4o for our classification task has significantly improved *Precision*, but, broadly speaking, has also reduced

Table 3.7: Accuracy results for scarce concepts.

Label	Precision	Recall
Colour	0.55	1
Firmness	1	0.66
Pathogen	0.44	0.66
Water Content	1	1

Recall considerably. Although the gain in *Precision* observed in our experiments is more than twice the loss in *Recall*, the latter often holds greater importance in requirements engineering, where filtering out false positives is generally an acceptable trade-off if it helps to avoid missing relevant labels. Another crucial factor to consider with fine-tuning is the cost of creating a sufficiently representative fine-tuning dataset. In real-world applications, this process can be costly, which may influence the decision on whether fine-tuning is worthwhile.

The above said, it is critical to note that our few-shot scenario represents a "best case" for few-shot learning. This is because we benefited from hindsight gained during our GT study, with the luxury of reflecting on numerous examples from that study and selecting a handful of the most comprehensive and inclusive ones for our few-shot prompt, ensuring the broadest possible coverage of the concepts. The question remains whether, in a domain where analysts are less familiar with the subject matter and lack the ability to choose from a large pool of examples as we did, they can still identify effective examples for few-shot learning that capture all the key aspects. Our current empirical evidence does not provide an answer to this question, as addressing it would require a separate line of investigation, such as user studies or controlled experiments using examples provided by non-authors for few-shot learning.

Handling scarce concepts. As mentioned in Section 3.6.5, certain concepts in our content model (Fig. 3.4), specifically *Colour*, *Firmness*, *Pathogen*, and *Water Content*, are scarce. For these concepts, BERT performs poorly, with both *Precision* and *Recall* falling below 5%. GPT-3.5 and GPT-4 show progressively better results than BERT; however, accuracy still remains low (detailed results can be found in our online material [102]). To improve accuracy for scarce concepts, we employ keyword-based classification, as discussed in Section 3.5.3. Since keyword-based classification introduces no random variation, we have excluded the results for these scarce concepts from the plots in Figs. 3.8 and 3.9. Instead, the results for the scarce concepts are presented in Table 3.7.

Comparing accuracy over Canadian vs. US regulations. Our LLM-based classification pipeline has comparable accuracy for both Canadian (FSRG) and US (FDA) regulations. In Table 3.8, we present the average (Avg) and standard deviation (SD) of accuracy metrics, calculated over 20 runs of the pipeline when instantiated with (fine-tuned) BERT, GPT-3.5, or GPT-4. The non-parametric pairwise Wilcoxon rank-sum test, per-

Table 3.8: Accuracy for Canadian (FSRG) vs. US (FDA) regulations.

		Precision		Recall	
		Avg	**SD**	**Avg**	**SD**
BERT	FSRG	0.87	0.02	0.87	0.03
	FDA	0.88	0.03	0.85	0.07
GPT-3.5	FSRG	0.88	0.02	0.84	0.03
	FDA	**0.90**	0.02	0.82	0.06
GPT-4	FSRG	0.88	0.02	**0.88**	0.04
	FDA	**0.90**	0.02	0.86	0.04

formed on the results of the individual LLMs, indicates no statistically significant difference between FSRG and FDA in terms of *Recall* (at the 5% confidence level). For *Precision*, there is no statistically significant difference in BERT results, but GPT-3.5 and GPT-4 perform statistically significantly better over FDA. These findings suggest that our pipeline performs equally well, if not better, on US regulations, despite the customization data being exclusively derived from Canadian regulations. Further details on our statistical tests can be found in our online material [102].

*The answer to **RQ2** is: Among fine-tuned models, GPT-4 outperformed GPT-3.5 and BERT, achieving ≈89%* Precision *and ≈87%* Recall *for our classification task. Comparing GPT-4 with few-shot learning against its fine-tuned version revealed a trade-off: few-shot learning increased* Recall *to ≈97% but decreased* Precision *to ≈65%. Our few-shot learning results may nonetheless be overoptimistic due to carefully selected examples. Despite being developed exclusively with Canadian regulations, our classification pipeline achieved comparable or better accuracy on selected US regulations, suggesting a degree of generalization across regulatory jurisdictions.*

RQ3: How do LLM-based classification models compare to simpler baselines in the context of food-safety regulations?

Figures 3.10 and 3.11 present boxplot results for 20 runs of the BiLSTM and keyword-search baselines. In the last four rows of Table 3.6 (shown earlier), we compare the accuracy of our best-performing models from RQ2 with that of these baselines.

The results indicate that BERT base and GPT-4o consistently outperform the baselines with large effect sizes in terms of *Recall*.

As for *Precision*, with one exception, BERT base either outperforms the baselines with statistical significance or achieves better results without statistical significance on the *Overall* and all L1 concepts. The exception is the *Measurement* concept, where BiLSTM yields a ≈3% better average *Precision*, albeit without statistical significance. For *Measurement*, BERT's *Recall* is ≈25% better. The 3% deficit in *Precision* is thus a reasonable trade-off

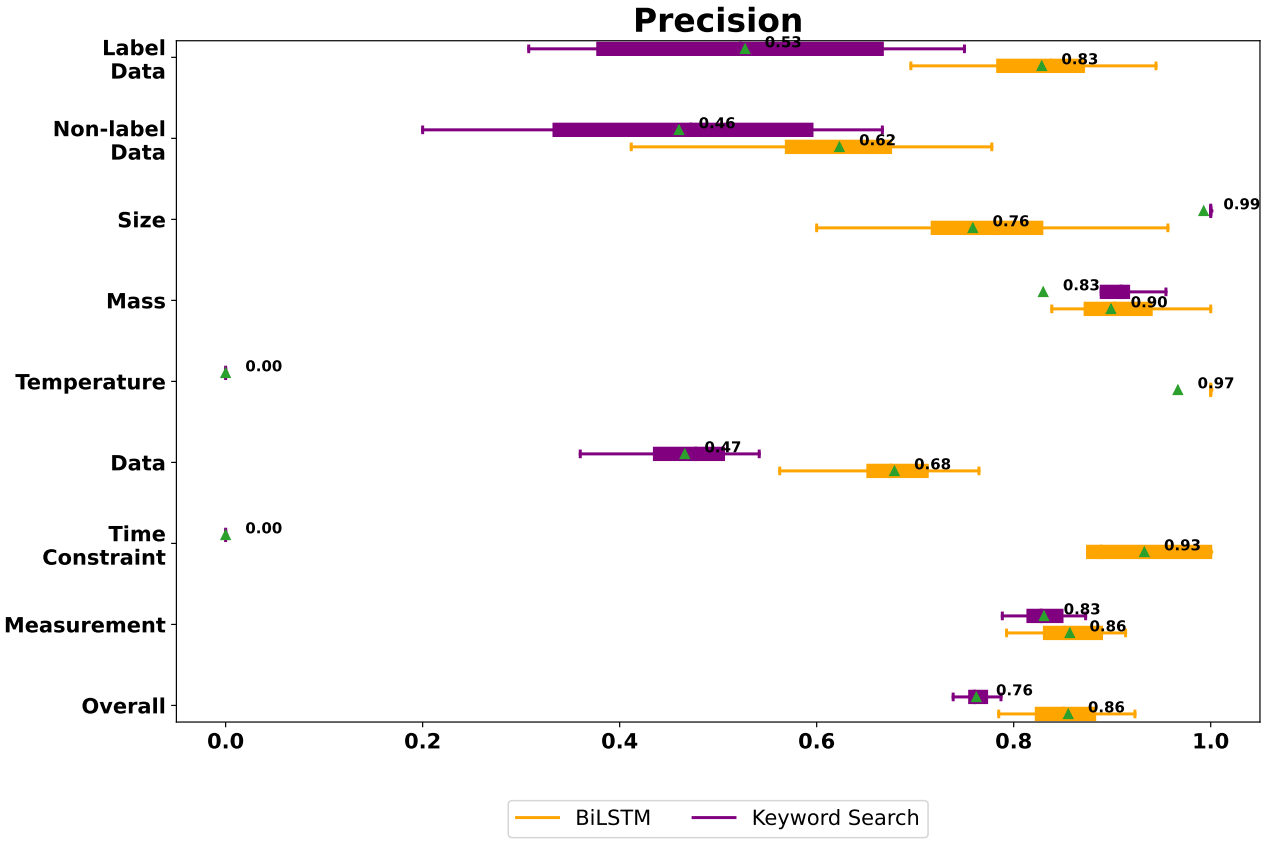


Figure 3.10: *Precision* for BiLSTM and Keyword Search (the x-axis represents the *Precision* values).

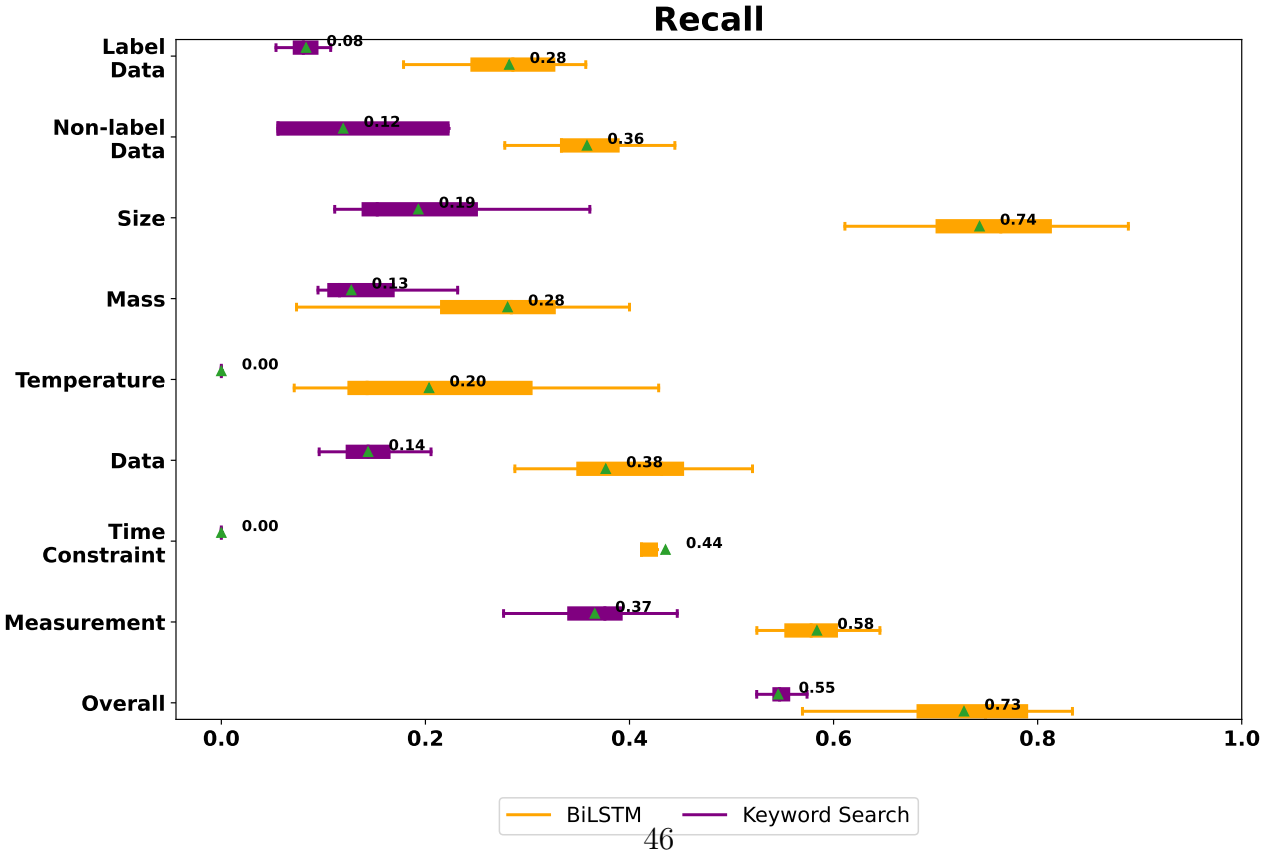


Figure 3.11: *Recall* for BiLSTM and Keyword Search (the x-axis represents the *Recall* values).

for the much better *Recall*. The statistical tests for L2 concepts, provided online [102], reveal a similar trend: BERT base is consistently better in terms of *Recall* and never at a major disadvantage in terms of *Precision*, given its considerably better *Recall*.

In a similar vein, GPT-4o consistently outperforms the baselines with large effect sizes in terms of *Precision* on the *Overall* and across all L1 concepts. The statistical significance tests for L2 concepts, provided online [102], reveal a similar trend with two exceptions: *Label Data* has a *Precision* of ≈79% (GPT-4o) vs. ≈83% (BiLSTM), and *Temperature* has a *Precision* of ≈87% (GPT-4o) vs. ≈97% (BiLSTM), with the difference in *Temperature* being statistically significant. However, the statistically significant gains in *Recall* – ≈41% and ≈69%, respectively, for these two concepts – favour GPT-4o, making it the better choice despite the (minor) deficits in *Precision*.

The answer to **RQ3** *is: For our classification task, BERT and GPT consistently outperform simpler, non-Transformer baselines in terms of* Recall*. Moreover, BERT and GPT have no significant disadvantage in* Precision*, with any* Precision *loss offset by much larger gains in* Recall*.*

3.6.6 Limitations and Validity Considerations

In this section, we discuss limitations and threats to validity. The validity considerations most pertinent to our empirical evaluation are *internal validity*, *external validity*, *construct validity*, and *statistical conclusion validity*.

Limitations. An important limitation of our evaluation is that our classification pipeline has not yet been applied in an industrial context. While, as mentioned in Section 3.1, our research was driven by a need identified by an industry partner, we have not yet reached the stage of deploying our pipeline in a full-scale industrial use case, such as for formulating the regulatory requirements of an actual food-safety monitoring system. As a result, we lack empirical evidence to evaluate the potential benefits of automating the classification of food-safety regulations in the specific industrial context that motivated our work.

Internal validity. The main threat to internal validity is bias. We implemented several measures to mitigate bias risks. First, the candidate and main supervisor jointly examined all provisions in our fine-tuning/training set and discussed challenging cases. Second, we delegated the task of annotating our test data entirely to third-party annotators, who had no knowledge of our automated classification solution. Finally, we measured inter-rater agreement over 10% of the test data to assess the reliability of the third-party annotators' work.

External validity. Our evaluation focuses exclusively on food-safety regulations in Canada, a specific legal domain within one country. Consequently, our findings may have limited broader applicability. However, it is important to recognize that food safety is one of the most far-reaching regulatory frameworks, given its direct impact on public health and well-being worldwide. Furthermore, as noted in Section 3.2, Canada ranks highly for "Quality

and Safety" in the Global Food Security Index, indicating that its food-safety regulations provide a good foundation for studying this area.

Another key consideration here is that regulatory requirements analysis has traditionally been domain-specific, as this approach yields more focused and concrete outcomes by addressing challenges unique to each domain [36, 73, 202]. We follow this practice, but acknowledge that it inherently imposes limitations on generalizability. While further case studies remain essential, we observe that our classification pipeline consistently performs well on test data from both Canada and the US. This finding lends confidence to the external validity of our results, especially within comparable regulatory environments.

Construct validity. Our quantitative analysis of classification accuracy followed established best practices, employing the standard metrics of *Precision* and *Recall*. As explained in Section 3.5, we chose individual sentences as the units of classification. This choice is supported by existing research on the classification of legal texts and considering the structural and conceptual parallels between food-safety regulations and other regulatory frameworks, such as data protection and privacy laws, examined in prior studies.

Statistical conclusion validity. To compare the accuracy of different LLMs, we used the Wilcoxon Rank-Sum Test. This non-parametric test enables us to compare the distributions of two independent samples – in our context, the distributions of *Precision* or *Recall* values from pairs of models, e.g., BERT base and GPT-4o. This test is well-suited to our analysis because our data – *Precision* or *Recall* values from 20 runs of each alternative solution – cannot be assumed to follow a normal distribution. Our use of the Wilcoxon Rank-Sum Test aligns with guidelines for evaluating randomized algorithms in software engineering [21]. We note that when comparing multiple alternatives – in our context, multiple LLMs – some researchers recommend first applying the Kruskal–Wallis Test to determine whether any overall significant differences exist among the alternatives before proceeding with Wilcoxon Rank-Sum Tests between pairs of alternatives. However, because we only had a small number of pairwise comparisons in our work, and we were specifically interested in direct comparisons between pairs of alternatives, we opted to bypass the Kruskal–Wallis Test and proceed directly with Wilcoxon Rank-Sum Tests. Conducting multiple statistical tests is known to increase the risk of Type I error inflation. Nonetheless, in accordance with the recommendations of Arcuri and Briand [21], we have chosen not to apply any corrections (e.g., Bonferroni correction) to control the family-wise error rate. This decision is particularly justified given the relatively small number of pairwise comparisons in our study.

We considered alternative tests, such as the Wilcoxon Signed-Rank Test, McNemar's Test, and the 5x2 cross-validation F-test, but favoured the Wilcoxon Rank-Sum Test for methodological reasons. The Wilcoxon Signed-Rank Test requires *multiple* datasets. Since we have only a single dataset (test set), applying this statistical test would have required splitting our dataset into smaller subsets. We concluded that this approach would not be methodologically sound, as it would artificially create multiple datasets from a single one, which was collected within the same domain. McNemar's Test also proved unsuitable, as it is primarily designed for nominal data and does not allow for direct comparison of our metrics, which are numeric. Finally, the 5x2 cross-validation F-test was ruled out to

avoid introducing bias through the mixing of training and test sets. Given the effort put into constructing a separate test set with third-party annotators, mixing this data with the training data – noting that the training data was annotated by the candidate and supervisors– would have conflicted with our objective of mitigating potential author bias.

Despite these precautions, we acknowledge that relying on a single test set limits the robustness of our statistical findings. Future studies with multiple test sets from diverse domains are needed to further increase robustness.

3.7 Discussion

This section provides practical insights into the implications of the empirical results presented in Section 3.6.

In our application context, LLMs tend to outperform simpler models. Our comparison with baseline models shows that, for our classification task, BERT and GPT are clearly more accurate than simpler models, such as LSTM and feature-based learning (specifically, Random Forest) when applied to keywords. This increased accuracy can be attributed to several factors, including the LLMs' ability to better understand context and meaning through pre-training, their stronger capacity to generalize across tasks, and their attention mechanisms, which help manage long-range dependencies more effectively than models like LSTM.

While it is generally unsurprising that LLMs outperform simpler models, providing evidence to justify their use for a specific task is still important. LLMs require considerably more computational power for training, fine-tuning, and querying, resulting in higher costs and a greater environmental impact. Therefore, striking a balance between accuracy and complexity – following the "Goldilocks principle" – is key to ensuring that the selected model is neither too simple nor too complex, but optimally suited for the task at hand.

Furthermore, it is important to recognize that not all LLMs or their variants perform equally well in every context. In our experiments, for instance, the specific Llama and Mixtral variants we tested did not consistently follow instructions, resulting in unpredictable outputs. This unpredictability made it challenging to use these models effectively, and consequently, conducting a systematic evaluation of their accuracy was not feasible.

***Using few-shot learning vs. fine-tuning could imply a trade-off between* Precision *and* Recall.** Emerging evidence suggests that out-of-the-box LLMs, unless customized to some extent, face challenges with domain-specific software engineering tasks [6]. In this chapter, we addressed one such task by focusing on food-safety regulations and adapting GT to assign meanings to concept labels. To classify accurately according to these labels, we needed to customize the LLM to capture the nuanced meanings and subtleties of the concepts identified in our GT study.

To customize an LLM, one can select between two main options: fine-tuning and few-shot learning, both of which were considered in RQ2 for GPT-4o. Our findings indicate that these two options lead to different trade-offs. Fine-tuning, which specializes a model by

adjusting its weights based on task-specific data, guides the model towards a detailed interpretation of concepts but potentially at the expense of narrowing its general understanding and making it overly confident in specific interpretations. Consequently, the model may overlook other valid variations and dismiss relevant instances that do not precisely align with the fine-tuning data, leading to a reduction in *Recall*. Few-shot learning, in contrast, maintains broader adaptability by presenting the LLM with only a few representative examples. This can result in more general understanding capabilities and, in turn, increase *Recall*. However, the flexibility may come at the expense of *Precision*, as few-shot learning may not provide enough guidance for the LLM to focus on important details necessary for accurately recognizing subtleties.

Resolving the above trade-off ultimately depends on whether the task prioritizes *Precision* or *Recall*. Although this trade-off may be inevitable, caution is needed when interpreting our current few-shot learning results. Specifically, based on our existing empirical data, we cannot determine whether comparable *Precision* levels can be achieved in a few-shot learning scenario where the provided examples are not as optimal. Nor can we ascertain whether adding more examples in few-shot learning could further increase *Precision* without compromising *Recall*. Addressing these questions requires substantial additional experimentation. However, based on our existing evidence, we can state that, for models like GPT-4o with extensive pre-training knowledge, few-shot learning shows promise as a more efficient and cost-effective alternative to fine-tuning for our classification task.

There is more to generative LLMs than classification accuracy. In our experiments with fine-tuned models, although GPT-4o slightly outperforms BERT, the practical advantage is marginal: GPT-4o shows only a 1% improvement in *Recall* (87% vs. 86%) and a 2% improvement in *Precision* (89% vs. 87%) on average. These results were obtained under identical fine-tuning conditions; thus, the effort spent on curating fine-tuning data was not a deciding factor. We conclude, therefore, that for our specific classification task, a fine-tuned BERT model performs nearly as accurately as a fine-tuned GPT model. Given BERT's significantly smaller computational footprint, the question arises whether using a resource-intensive model like GPT is justified for our classification task.

Our answer to this question is as follows: Classification is often an important but preliminary task for many follow-on tasks that analysts need to perform. Generative models like GPT offer additional capabilities, such as providing explanations, rationale, and the ability to follow conversational instructions. These features are key for stakeholders who require justification for decisions or wish to engage interactively in the classification process, correcting mistakes as they arise. The ability to request explanations or provide real-time feedback is absent in non-generative models like BERT. In our application context, the added value of generative models like GPT lies not necessarily in a drastic improvement in classification accuracy but in supporting more natural, human-in-the-loop interactions with legal texts in the future.

The classification pipeline's execution time is sufficient for practical use. We measured the execution times of our classification pipeline using different LLMs. Since our classification task is a one-time process, execution times do not impact the choice of which LLM to use, as long as the times remain reasonable. Below, we discuss the execution

times of our most accurate pipelines, which were created using BERT and GPT, noting that these times are not directly comparable because the models were deployed in different environments.

To measure BERT's execution time, we used a modest platform to replicate the resources available to end-users. Specifically, we used the free version of Google Colab Cloud with the following specifications: Intel Xeon CPU@2.30GHz, Tesla T4 GPU, and 13GB RAM. BERT's execution time includes a one-time pre-processing and a one-time loading phase, followed by label predictions in batches of 16. In our Colab setup, pre-processing and loading together took < 0.5 seconds. Label prediction with BERT base for one batch took $\approx$4.32 seconds, resulting in $\approx$0.27 seconds per provision. For GPT, we used OpenAI's token-based plan (paid subscription service). Our GPT experiments employed the same pre-processing as BERT, with the loading of the (fine-tuned) LLM managed by OpenAI. Calculating precise execution times for GPT is not feasible due to lack of control over OpenAI loads during execution as well as factors such as rate limits and OpenAI's safeguards against disruptive attacks. For both GPT-3.5-turbo and GPT-4o, the average time to complete one run over our entire test set was $\approx$5 minutes, yielding an average execution time of $\approx$0.7 seconds per provision. These results indicate that execution time is not a barrier to adoption, irrespective of the LLM being used.

Overall, the resource requirements for fine-tuning were modest and one-time. We believe this cost is justified by the performance gains observed—particularly the higher *Precision* achieved by fine-tuned models. This is especially relevant in the food-safety domain, where misclassification of regulatory provisions can result in significant compliance risks. While few-shot prompting offers convenience, it does so at the expense of precision. Thus, for applications where classification accuracy is critical, fine-tuning is a worthwhile investment.

3.8 Data Availability

Our full replication package is available online [97], including:

- Our dataset, keywords, and SFCR exclusions [101];
- The hyperparameters and dataframes for different models, along with complementary statistical analysis results [102];
- The implementation of all algorithms, including baselines and evaluation scripts [103].

Compliance with Ethical Standards

Ethical Approval. This research did not involve human participants or animals; therefore, ethical approval was not required.

Informed Consent. No personal data or identifiable information is reported in this research; informed consent is not applicable.

3.9 Conclusion

We presented two main contributions: (1) a Grounded Theory study that characterizes food-safety concepts in regulatory provisions impacting modern software-intensive food-safety systems, and (2) an LLM-based classification pipeline that classifies the provisions of food-safety regulations based on their relevance to systems and software requirements. We examined the effectiveness of our LLM-based classification pipeline by instantiating it with two families of LLMs, namely BERT and GPT. Fine-tuning GPT-4o yielded the highest accuracy, achieving 89% *Precision* and 87% *Recall*. When we implemented few-shot learning with GPT-4o, *Recall* improved to 97%, but *Precision* dropped to 65%. While our classification pipeline was tailored to Canadian regulations, it effectively processed provisions from both Canadian and US regulations. Finally, we observed that LLMs significantly outperformed simpler, non-Transformer-based baselines.

Although our experimental results do not directly measure practical usefulness, they indicate that the best-performing LLMs achieve high *Precision* and *Recall*, suggesting that automation is valuable for our classification task. Manual review of regulatory documents is known to be tedious and error-prone, especially under the time pressures common in industrial environments. In this context, even less-than-perfect automated results can be beneficial, provided that *Recall* remains consistently high – an outcome observed with the most accurate models in our evaluation.

As part of our work, we curated a dataset containing over a thousand labelled provisions. We are sharing this dataset with the research community to stimulate further research in the domain of food supply chain and safety. In future work, we plan to (1) study the direct derivation of requirements from food-safety regulations, (2) explore the generalizability of our food-safety concepts beyond North America, and (3) conduct user studies to evaluate the practical benefits of our automated classification solution.

Chapter 4

Rethinking Legal Compliance Automation: Opportunities with Large Language Models

As software-intensive systems face growing pressure to comply with laws and regulations, providing automated support for compliance analysis has become paramount. Despite advances in the Requirements Engineering (RE) community on legal compliance analysis, important obstacles remain in developing accurate and generalizable compliance automation solutions. This chapter[1] highlights some observed limitations of current approaches and examines how adopting new automation strategies that leverage LLMs can help address these shortcomings and open up fresh opportunities. Specifically, we argue that the examination of (textual) legal artifacts should, first, employ a broader context than sentences, which have widely been used as the units of analysis in past research. Second, the mode of analysis with legal artifacts needs to shift from classification and information extraction to more end-to-end strategies that are not only accurate but also capable of providing explanation and justification. We present a compliance analysis approach designed to address these limitations. We further outline our evaluation plan for the approach and provide preliminary evaluation results based on DPAs that must comply with the GDPR. Our initial findings suggest that our approach yields substantial accuracy improvements and, at the same time, provides justification for compliance decisions.

4.1 Introduction

Software-intensive systems are increasingly subject to regulatory mandates. Deriving legal requirements and assessing compliance for these systems requires the ability to accurately interpret and apply legal provisions, including identifying what pertains to software and

[1]This chapter was published at the 32nd IEEE International Requirements Engineering Conference (RE 2024) [109].

how. Due to the close relationship between regulatory texts and requirements, the Requirements Engineering (RE) community has extensively studied automated processing of legal texts using Natural Language Processing (NLP) and Machine Learning (ML) to facilitate the definition and analysis of legal requirements. Notable attempts in this direction include works such as [3, 10, 12, 13, 16, 34, 36, 37, 139, 194, 201, 202, 239, 240].

Breaux et al. [34, 36, 37] extract rights and obligations from regulatory texts. Kiyavitskaya et al. [139] provide a tool based on rights and obligations to analyze policy documents. Zeni et al. [239, 240] and Sleimi et al. [202] extract semantic metadata from legal texts. Sleimi et al. [201] further use semantic metadata to build a knowledge base that can be queried. Sannier et al. [194] automate detection and resolution of cross references. Amaral et al. [12] extract metadata from privacy policies and automate GDPR compliance checking. Abulhajia et al. [3] automate question answering of compliance requirements. Amaral et al. [10, 13] automate completeness checking of DPAs against GDPR. In addition, Anish et al. [16] classify security and privacy requirements from obligations in software engineering contracts.

Despite the significant progress made in the RE community for automated legal text processing, important limitations remain. Our goal in this chapter is to highlight salient existing limitations and discuss how a rethinking of current automation strategies will help us take advantage of the opportunities provided by LLMs [39].

4.1.1 Limitations in Existing Research

Reliance on sentences as units of analysis

Existing approaches mostly promote a sentence-by-sentence analysis of legal texts. We have observed three key issues caused by this decision that can significantly affect accuracy.

First, interpreting sentences often requires contextual understanding beyond the immediate sentence structure, as sentences can be related to previously mentioned categories or definitions. A notable example involves sentences with linguistic proforms like "these". Consider for example the following snippet from a DPA, introduced in Section 4.2.1: "Depending on the security classification, buildings [...] may be further protected by additional measures. These measures include specific access profiles, video surveillance, intruder alarm systems, and biometric access control systems." Here, the second sentence alone does not expressly indicate that the mentioned measures are about protecting buildings.

Second, it is common for a legal concept to be defined, with subsequent sentences drawing upon the definition. For instance, in a privacy policy, there may be a statement defining "personal data". Subsequent sentences then use "personal data" in various contexts. To illustrate, consider the following snippet: "By accepting this policy, you are providing personal data as defined below [...]: information you provide to us verbally, electronically, or in writing; [...] information obtained from third parties [...]; or information obtained through cookies. Your personal data might be disclosed to the tax authorities or other third parties [...]". To interpret what may be disclosed as per the provision in the second sentence, one needs to interpret it in view of the definition of "personal data" in the first one.

Third, legal texts are frequently interwoven with cross-references [194]. This practice adds complexity by requiring an understanding of the broader content across a collection of legal documents. For instance, consider the following privacy-policy excerpt: "45. (1) A health information custodian may disclose personal health information to [...], contingent upon the entity meeting the requirements outlined in subsection (3)". Here, fulfilling the requirements specified in the provision further depends on the content of subsection (3). The ability to interpret text within the context of cross-references is thus crucial for grasping the full meaning of a given provision.

Automation strategies lack justification for decisions and are either coarse-grained or entail significant manual effort to build

Automation for legal texts employs one of the following three strategies or a combination thereof. We argue that these strategies not only fail to offer justification and rationale for decisions – a capability now achievable with LLMs – but they either are too coarse-grained, potentially compromising accuracy, or require considerable manual effort, making compliance automation costly to implement.

Strategy 1: Extracting metadata or semantic frames and then executing predefined rules over the metadata/frames to check compliance. Examples of works employing this strategy include: (a) Xiang et al. [236], who propose a rule- and semantic role-based approach to verify privacy-policy completeness against GDPR; (b) Amaral et al. [10–12], who extract metadata and check for the presence of metadata types envisaged by compliance rules; and (c) Amaral et al. [13], who compare GDPR requirements and DPA sentences based on semantic frames for compliance-rule assessment.

Strategy 2: Projecting regulatory texts into an embedding space and then measuring the similarity between these texts against (textual) compliance rules or a labelled group of regulatory provisions. An example of research employing this strategy is that of Amaral et al. [10,12], who create sentence embeddings and utilize semantic similarity for completeness checking of privacy policies and DPAs.

Strategy 3: Using encoder-only single-input transformer models, such as Bidirectional Encoder Representations from Transformers (BERT) [52], to classify which compliance rules are satisfied by each legal text. An example of work applying this strategy is that of Ilyas et al. [127], who use BERT to construct a multi-label classification pipeline for identifying sentences in DPAs that meet specific compliance rules.

Limitations. A notable limitation of all the strategies outlined above is their inability to provide justification for compliance-rule satisfaction or violation. Despite the fact that generative LLMs may still lack sophisticated logical reasoning abilities, they are often capable of offering simple yet valuable explanations to clarify why a specific rule has been met or breached.

In addition to the above general limitation, the three existing strategies suffer from further drawbacks. For Strategy 1, these additional drawbacks are: (1) There is a significant amount of manual work involved, typically in the form of some kind of qualitative

study (e.g., grounded theory) to identify the rules and frames. (2) There is a challenge when metadata/rules are less prevalent in the dataset and are considered rare instances. In such cases, this strategy requires additional steps such as keyword search [12]. (3) Regulations are subject to continuous change, which can impact the compliance checking process. This necessitates regularly extracting more concepts or creating new rules, resulting in additional manual effort.

As for Strategy 2, the additional drawbacks include a lack of precision and unsuitability for many rules. As observed by Amaral et al. [13], semantic similarity is not always indicative of rule applicability due to inherent limitations in capturing the nuances of legal texts. Consider for example the following DPA excerpt (from [13]) : "This agreement is between Sefer University [...] the "Company"; and Levico Accounting GmbH [...] WHEREAS (A) The Company acts as a Data Controller. (B) Levico Accounting GmbH acts as a Data Processor. (C) The Processor wishes to lay down their rights and obligations." Further, consider compliance rules R1 and R2, which we would like to assess for satisfaction: R1: "The DPA shall contain at least one controller's identity and contact details."; R2: "The DPA shall contain at least one processor's identity and contact details." In this example, although the excerpt satisfies R1 and R2, the semantic similarity, as computed by cosine similarity, would be low. This issue can be partially rectified by enhancements such as bi-encoder fine-tuning [187]. Nevertheless, the coarse-grained semantic similarity measures commonly used in Strategy 2 still tend to overlook the subtleties in legal texts, even after fine-tuning.

Finally, with regard to Strategy 3, the main additional limitation is that this strategy does not offer the flexibility to differentiate between partial and full compliance. To illustrate, consider the following DPA excerpt (borrowed from Amaral et al. [10]): "A description of the nature of the Personal Data Breach, including, if possible, the categories and the approximate number of affected Data Subjects and the categories and the approximate number of affected registrations of personal data." Further, consider the following multi-part compliance rule: "The notification of personal data breach shall at least include (a) the nature of personal data breach; (b) the name and contact details of the data protection officer; (c) the consequences of the breach; (d) the measures taken or proposed to mitigate its effects." In this example, the DPA excerpt satisfies some but not all parts of the multi-part rule. In contrast, querying a generative LLM such as ChatGPT would yield more precise answers, such as the following: *"No, the sentence [DPA snippet] does not fully satisfy the rule as it only addresses one part of the required elements."* Another limitation of Strategy 3 (which is shared with limitations of Strategy 1) is the handling of rare cases. Addressing this issue requires either expanding the dataset, as suggested by Ilyas et al. [127], or disregarding rare cases.

4.1.2 Contributions

We present an approach and preliminary results for addressing legal compliance automation in a more end-to-end manner. Our proposed approach has been designed to eliminate the need for explicit specification of metadata, frames, and rules and to provide a finer-grained, semantics-based analysis compared to approaches based solely on embeddings.

Using LLMs such as Generative Pre-trained Transformer (GPT-3.5) [173] and newer models, which support context windows of up to 128k tokens (in contrast to BERT's limited 512 tokens [52]), enables us to consider a much broader context for compliance automation. Furthermore, the generative nature of these LLMs supports prompt-based interactions to provide rationalization and justification for satisfaction or violation of compliance rules. Another key advantage is the few-shot learning capabilities of these newer LLMs, which reduce the need for extensive data labelling and training – a common requirement in previous research.

To evaluate our proposed approach, we perform a preliminary assessment using DPAs (as previously illustrated in this section and to be explained further in Section 4.2.1) as the legal artifacts of interest. Our main aim is to examine the accuracy of various generative LLMs in distinguishing compliance and non-compliance. This evaluation focuses on understanding the impact of broader context, transitioning from individual sentences to entire paragraphs, while also considering implications regarding cost and time.

From our experiments with several LLMs, we make the following preliminary observations: (1) There are considerable variations across different LLMs in terms of accuracy. The choice of LLM is thus likely to be a critical factor when approaching legal compliance automation. (2) The ability to account for a larger context when interpreting a given regulatory provision leads to substantial improvement gains, as high as 40% in our experiments.

Structure. Section 4.2 presents background. Section 4.3 compares with related work. Section 4.4 describes our approach. Section 4.5 offers implementation details. Section 4.6 plans the approach evaluation. Section 4.7 outlines our preliminary evaluation results. Section 4.9 lays out future research. Section 4.8 provides links to our online data and implementation [100].

4.2 Background

In this section, we provide the necessary legal and technical background for our approach.

4.2.1 General Data Protection Regulation (GDPR)

GDPR [114] is a privacy and data protection law enacted by the European Union to regulate the processing of personal data and enhance individuals' control over their information. Under GDPR, an organization is identified as either a data controller or a data processor. Data controllers are responsible for determining the purposes and mechanisms of collecting and processing personal data, while data processors process the data as per the controller's directives. In this chapter, we use DPAs, as defined by GDPR, to illustrate ideas and conduct a preliminary evaluation. DPAs are binding agreements that specify the responsibilities and rights of controllers and processors. Most of the content in DPAs is software-related, making them of direct interest to the RE community [10]. For GDPR compliance, DPAs must follow GDPR Article 28.

4.2.2 Large Language Models (LLMs)

We use LLMs to assess the compliance of legal documents. LLMs are statistical models trained to predict and generate coherent, contextually relevant text. These models are broadly categorized into *generative models*, such as GPT [185], which excel in text production, and *discriminative models*, such as BERT [52], which are optimized for classification and analysis tasks. LLMs undergo extensive pre-training on vast text corpora, enabling them to grasp language nuances, contextual information, and complex linguistic patterns.

Users can guide LLMs by providing *prompts*, which are specific instructions or queries that guide the model to generate relevant and contextually appropriate responses [94]. Various prompting strategies exist for LLMs [146], such as zero-shot, few-shot, chain-of-thought, and tree-of-thought prompting. In this chapter, we explore zero-shot prompting, where the LLM generates responses without prior specific examples, relying solely on the task's context and instructions.

To optimize their performance for specific tasks, such as compliance checking, LLMs are *fine-tuned* – a process that involves adjusting the pre-trained models to suit particular domain requirements [224]. For example, fine-tuning an LLM on GDPR-compliant documents can enhance the LLM's accuracy in interpreting and applying GDPR provisions.

Our preliminary experimentation in this chapter encompasses three LLMs: BERT [52], GPT [185], and Mixtral [121]. BERT uses bidirectional training of transformers – a popular attention model for language modelling. BERT comes in two sizes: BERT base and BERT large. BERT large can be more accurate for certain language understanding tasks but is computationally more expensive. BERT can differentiate between capitalized and non-capitalized text. With BERT uncased, the text is first converted to lowercase before tokenization, whereas with BERT cased, the tokenized text remains the same as the input text with respect to capitalization. Previous RE research favours the cased model for analyzing requirements specification and legal documents. For our experimentation with BERT, we utilize BERT base cased. GPT is a transformer-based family of models, pre-trained to predict the next token in a sequence. GPT has shown the ability to capture subtle linguistic patterns and excel in many language understanding and generation tasks. We use GPT-3.5-turbo and GPT-4-turbo, more specifically gpt-3.5-turbo-0125 and gpt-4-0125-preview, for our experiments. Mixtral is a specialized LLM that integrates multiple linguistic capabilities, including translation, summarization, and question answering. It employs a hybrid architecture, combining traditional rule-based approaches with ML to achieve accurate language understanding and generation. We use Mixtral-8x7B-Instruct-v0.1 for experimentation.

4.3 Related Work

Significant progress has been made in completeness and compliance checking of legal requirements and software-related regulatory artifacts through the application of traditional machine learning algorithms and LLMs. In this section, we compare our approach with existing methods, emphasizing differences in methodology and the limitations of prior work.

Among regulatory artifacts relevant to software, privacy policies have arguably received the most attention. Traditional ML-based approaches for completeness and compliance checking of privacy policies have largely relied on datasets or conceptual frameworks proposed by Wilson et al. [232], Torre et al. [215], Liu et al. [147], and Amaral et al. [11].

Some previous studies link privacy policies to software code. Fan et al. [61] use traditional ML classifiers to evaluate GDPR compliance in mobile health applications, while Hamdani et al. [93] combine ML and rule-based methods for automated compliance checking of privacy policies with GDPR. Xie et al. [238] employ Bayesian classifiers and NLP to examine the compliance of privacy policies in virtual personal assistant applications. Amaral et al. [12] combine NLP and feature-based learning to extract metadata from privacy policies and automate GDPR compliance checking. These works focus exclusively on privacy policies. Furthermore, in terms of an automation strategy, they pursue one or a combination of the following: (1) metadata extraction, (2) semantic-similarity comparison, and (3) hand-crafted rules; these strategies pose problems as discussed in Section 4.1.1 under Strategies 1 and 2.

Harkous et al. [95], Mousavi et al. [165], and Tang et al. [210] respectively use Convolutional Neural Networks, BERT, and GPT for privacy-policy analysis. Aside from, again, focusing exclusively on privacy policies, these works remain premised on metadata extraction, thus suffering from the limitation discussed under Strategy 1 in Section 4.1.1. In the case of Mousavi et al. [165] and Tang et al. [210], which are respectively using BERT and GPT, we observe that both use LLMs in a discriminative mode. BERT lacks a decoder-based architecture that can be prompted for generative tasks. As for Tang et al. [210], they do not exploit the generative capabilities of GPT and instead choose to use it for classification in a way similar to more traditional approaches.

Another software-related regulatory artifact that has been studied more recently are DPAs (introduced in Section 4.2.1). Amaral et al. [10] combine traditional ML classifiers with cosine similarity-based classification to assess the completeness of DPAs against GDPR. This involves checking each sentence embedding from DPA against all embeddings for all rules. More recently, Amaral et al. [13] have introduced an NLP-based approach using semantic frames to check DPA compliance with GDPR. Semantic frames are extracted from both the sentences in the DPAs and the GDPR rules, followed by a comparison to identify similarities and discrepancies between the two sets of frames. These works (1) are sentence-level, (2) rely on metadata extraction, and (3) compare semantic similarity, or use semantic frames and rule-based methods that have issues related to Strategies 1 and 2. In contrast, our approach exploits LLM capabilities, aiming to reduce the dependency on feature-based learning and semantic similarity.

Ilyas et al. [127] explore multiple AI solutions for DPA compliance checking, including traditional ML classifiers, Bidirectional Long Short-Term Memory (BiLSTM), and BERT. This work has been applied to a specific set of rules, and excludes several regulatory rules such as metadata and controller rights which earlier work by Amaral et al. [10] addresses, most likely due to lack of sufficient data or the recognition that these requirements are difficult to analyze when the context window is limited to individual sentences. This approach overlooks the broader textual context of the input sentences from the input text

and treats rules merely as labels. In contrast, our proposed approach utilizes prompting strategies made possible by generative LLMs, enabling us to reason about compliance within a context broader than individual sentences and also alleviating the need to manually encode compliance requirements into formally specified rules.

The idea of broadening the context for the analysis of legal provisions can, in principle, be generalized beyond paragraphs. For example, Sun et al. [209] and Chen et al. [45] propose approaches for handling long documents. The former approach works by decomposing queries into a sequence of actionable tasks, structuring interaction with the document through a systematic plan and execution process. The latter approach creates a summary-based tree structure from long texts, enabling the model to navigate and retrieve information efficiently in response to specific queries. These works raise the prospect that legal documents can be analyzed in their entirety for legal compliance. Unfortunately, as of this writing, these techniques fall short in the legal domain, where it is essential to preserve the details of the regulatory text and fully grasp the legal implications. The decomposition of queries into actionable tasks as proposed by Sun et al. [209] could lead to an oversimplification of legal texts, thereby compromising the integrity and depth of the interpretation necessary for effective compliance checking. In a similar manner, the summary-based approach of Chen et al. [45] risks losing essential legal nuances, noting that legal documents feature interconnected information, where precision of wording and context is paramount.

We note that our research is not the first to recognize the potential of LLMs for compliance automation and inconsistency/violation detection. Berger et al. [28] propose a combination of retrieval and LLM-based zero-shot learning to automate compliance checking of audit decisions, while Fantechi et al. [62] introduce an LLM-based approach for identifying inconsistencies in natural-language requirements. Although this latter approach does not explicitly address legal compliance, we share some common elements with it. The main distinction between our research and the recent studies mentioned above lies in our interest in creating appropriate units of analysis for legal automation. In doing so, our research aims to explore the inclusion of cross-referenced content as a way to assemble a self-contained context, with just the right amount of information to support answering specific questions about compliance and non-compliance.

Santos et al. [196] frame requirements analysis as a satisfiability problem solvable through in-context learning with LLMs. They formulate of the "satisfaction argument" $(S, D \vdash R)$ as a Natural Language Inference (NLI) task. The model reasons over a premise (composed of system specifications and domain knowledge) to determine if it logically entails a hypothesis (the requirement). Our work is closely aligned with this perspective, but instantiates it within the domain of legal compliance. In our work, the premise consists of paragraph-level passages from regulatory artifacts (e.g., DPAs) while the hypothesis corresponds to an explicit regulatory compliance rule. Beyond RE, recent work has proposed broad benchmarks to measure legal reasoning capabilities of LLMs. In particular, Guha et al., [91] provide a collaboratively constructed suite of legal reasoning tasks, including tasks that resemble entailment-style reasoning over contracts (e.g., contract NLI). This benchmark highlights the diversity of legal reasoning behaviors that LLMs must support; our work complements this benchmarking perspective by focusing on end-to-end compli-

ance judgments over realistic regulatory artifacts with explicit rule grounding and context construction.

Also there are platforms operationalizing governance, risk, and compliance that manage control libraries, evidence collection, and audit workflows (e.g., IBM OpenPages [126], ServiceNow GRC [198], and Archer [20]). In adjacent spaces, privacy and AI governance platforms (e.g., OneTrust [170]) support risk oversight and the production of audit-ready documentation. In software delivery pipelines, policy-as-code engines such as Open Policy Agent (OPA) [172] enable automated enforcement of machine-checkable policies over system configurations and artifacts. These platforms are complementary to our study: they provide robust workflow, evidence, and accountability mechanisms, but typically assume that regulatory obligations have already been translated into internal controls, policies, or checklists.

4.4 Approach

Figure 4.1 presents an overview of our approach, which uses a systematic method employing LLMs to evaluate regulatory artifacts for compliance with specific regulations. The approach consists of three steps. The first step (❶) is to create passages from the regulatory artifact that the user wants to verify for compliance, e.g., a DPA. The second step (❷) is to generate a prompt for LLMs. This step takes as input the passages generated in the first step, along with the compliance rules to be verified (e.g., from the GDPR), and a configurable prompt template. The third step (❸) is to present the prompt generated in the second step to LLMs to draw inferences about compliance and non-compliance. The output of the approach is a compliance report, accompanied by an explanation and justification for each determination.

Step 1) Content Chunking. The regulatory artifact subject to compliance checking is fed to the pipeline. The pipeline segments the regulatory artifact into manageable content chunks, primarily paragraphs (Fig. 4.1, Step ❶). This process ensures that the chunks, once incorporated into the overall prompt, will fit within the token limit of the LLMs used. A paragraph is explicitly defined as a textual segment initiated by a top-level numbered heading and continuing until the next such heading or the end of the document. This is implemented using the pattern `r"(?=\d+\.\s*.+)(.*?)(?=(?=\n\d+\.\s*)|\Z)"`. This pattern identifies paragraphs by detecting lines that begin with a top-level heading, i.e., a number followed by a period and n optional spaces, followed by the heading title. It uses a lookahead to stop the capture right before the next top-level heading or the end of the document. If a full paragraph were to exceed the token limit, it would need to be either truncated or summarized. However, in our investigation of LLMs, we did not encounter such situations due to their reasonably large token limit. The output of this step is a set of passages each within the LLM's token limit, ready for prompt construction. Figure 4.2 shows a snippet of a DPA as the regulatory artifact subject to GDPR (from Amaral et al. [10]), along with examples of passages for both the sentence level (❶) and the paragraph level (❷).

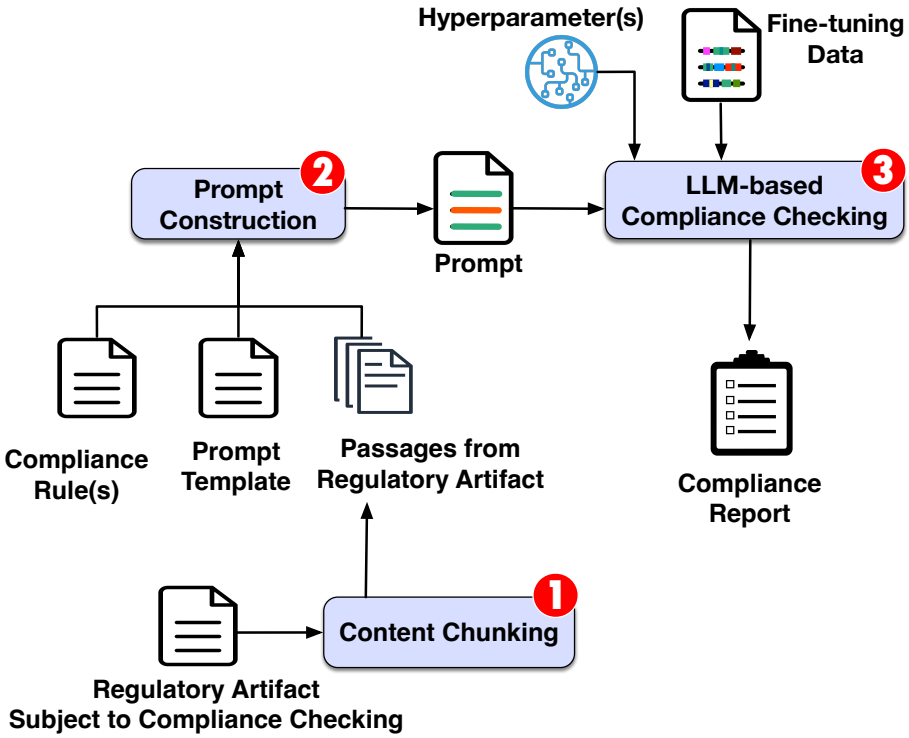


Figure 4.1: Approach overview.

Step 2) Prompt Construction. Our approach constructs tailored prompts to guide the LLMs in eliciting rule-specific responses, based on the input compliance rules. Prompts, designated to extract both the rule applicability and the LLMs' explanation and justification, consist of three parts: *Prompt Template*, *Compliance Rules*, and *Passages from Regulatory Artifact* (Fig. 4.1, Step ❷).

For instance, Fig. 4.2 (bottom-right) shows a (trimmed) list of 46 regulatory rules for evaluating DPAs. We denote rules as R_i along with their definitions. In our example, i ranges from 1 to 46, with 99 being a sentinel value indicating non-applicability. In Fig. 4.2, we illustrate that for sentence-level analysis (❶), the Prompt Template differs slightly from that used for paragraph-level analysis (❷). Specifically, in sentence-level analysis, the model is provided with a single sentence and asked to make predictions based solely on that sentence. In contrast, in paragraph-level analysis, the model is tasked with making inferences over the input text within the context of the entire paragraph. This difference enhances the model's ability to effectively infer meaning based on the given passage.

The prompt is presented in a structured chat format for the LLMs, specifying roles for system instructions, user inquiries, and the assistant's responses, following best practices where applicable [115, 175]. The system role provides instructions for the model to follow (detailed in the Prompt Template in Fig. 4.2). The user role presents input that the model should respond to (detailed in the Passages from Regulatory Artifact in Fig. 4.2). The assistant role represents the model's response to the user's input (detailed in the

(a) Data Processing Agreement:

…1 [**9. Duration**]
2 [9.1 The Data processor Agreement shall remain in force until the Main Agreement is terminated.]

3 [**10. Termination**]
4 [**10.1** The Data Processor's authorization to process Personal Data on behalf of the Data Controller shall be annulled at the termination of this Data Processor Agreement.]
5 [**10.2** The Data Processor shall continue to process the Personal Data for up to three months after the termination of the Data Processor Agreement to the extent it is necessary and required under the Applicable Law.]
6 [In the same period, the Data Processor is entitled to include the Personal Data in the Data Processor's backup.]
7 [The Data Processor's processing of the Data Controller's Personal Data in the three months after the termination of this Data Processor Agreement shall be considered as being in accordance with the Instruction.]
8 [**10.3** At the termination of this Data Processor Agreement, the Data Processor and its Sub- Processors shall return the Personal Data processed under this Data Processor Agreement to the Data Controller, provided that the Data Controller is not already in possession of the Personal Data.]
9 [The Data Processor is hereafter obliged to delete all the Personal Data and provide documentation for such deletion to the Data Controller.]

10 [**11. Contact**]
11 [**1.1** The contact information for the Data Processor and the Data Controller is provided in the Main Agreement.]

12 [**Sub-appendix A**]

13 [**1. Personal Data**]
14 [**1.1** The Data Processor processes the following types of Personal Data in connection with its delivery of the Main Services:]

15 [(i) Ordinary contact information on relevant employees from the Data Controller.]
16 [(ii) Users of the Main Services: names, telephone numbers, e-mails and user type.]
17 [(iii) Personal data provided by the users in connection with their use of the Main Services (these personal data are not seen or accessed by the Data Processor unless the Data Processor after the request hereof from the Data Controller assists with support and bug fixing).]

18 [**2. Categories of data subjects**]
19 [**2.1** The Data Processor processes Personal Data about the following categories of data subjects on behalf of the Data Controller:]

20 [(i) Customers]
21 [(ii) End-users]…

(b) Prompt:

❶ **Sentence-level Passage:**

[(i) Ordinary contact information on relevant employees from the Data Controller.]

❷ **Paragraph-level Passage:**

[**1.** Personal Data]
[**1.1** The Data Processor processes the following types of Personal Data in connection with its delivery of the Main Services:]
[(i) Ordinary contact information on relevant employees from the Data Controller.]
[(ii) Users of the Main Services: names, telephone numbers, e-mails and user type.]
[(iii) Personal data provided by the users in connection with their use of the Main Services (these personal data are not seen or accessed by the Data Processor unless the Data Processor after the request hereof from the Data Controller assists with support and bug fixing).]

Prompt Template:

❶
You are a legal expert trained to identify applicable {Compliance Rules} based on a given {text}.
Your response should only include the rule identifier (e.g., 'R5') if applicable. If there is no direct connection to any Compliance Rule, respond with 'R99'.
Do not include any explanations or additional text.
Follow this format strictly.
Then, provide your rationale for the decision.

❷
You are a legal expert trained to identify applicable {Compliance Rules} based on a given {text} within its specific {context}. When provided with the {text} and its {context}, your response should only include the rule identifier (e.g., 'R5') if applicable. If there is no direct connection to any Compliance Rule within the context provided, respond with 'R99'. Follow this format strictly.
Then, provide your rationale for the decision.

Compliance Rules:

R1 - The DPA shall contain at least one controller's identity and contact details.
R2 - The DPA shall contain at least one processor's identity and contact details.
R3 - The DPA shall contain the duration of the processing.
…
R16 - The processor shall take all measures required pursuant to Article 32 or to ensure the security of processing.
R17 - The processor shall assist the controller in fulfilling its obligation to respond to requests for exercising the data subject's rights.
…
R46 - Any controller involved in processing shall be liable for the damage caused by processing which infringes the GDPR.

Figure 4.2: (a) Illustrative data processing agreement (DPA), (b) Prompt including three parts: Passages from Regulatory Artifact (sentence-level (❶) or paragraph-level (❷) input), Prompt Template, and Compliance Rules.

1

R99
Based on the information provided, it is not possible to determine whether the sentence "Ordinary contact information on employees from the Data Controller" satisfies rule R5 - The DPA shall contain the types of personal data. To assess compliance with this rule, we would need to review the entire Data Processing Agreement (DPA) and determine if it includes a clear and comprehensive list of the types of personal data that will be processed.

2

R5
The sentence "Ordinary contact information on employees from the Data Controller" satisfies the rule "R5 -The DPA shall contain the types of personal data".
It clearly states that the Data Processor processes ordinary contact information on relevant employees from the Data Controller, which falls under the category of personal data.

Figure 4.3: Illustrative compliance report generated by GPT-4 Turbo: for sentence-level inputs (❶) without context, and for paragraph-level inputs (❷) with context.

Compliance Report in Fig. 4.3).

Additional examples demonstrating different input texts at both the sentence and paragraph levels, along with the model's explanation and justification, are available in our online repository [100]. The corresponding implementation (see Section 4.5) is also provided in the code section of the repository [99].

Step 3) LLM-based Compliance Checking. Once tailored prompts are constructed, we employ zero-shot learning, where the model generates responses (Fig. 4.3) without prior training on similar tasks, based solely on the instructions provided in the prompt (Fig. 4.2). In the future, this process may be enhanced with (optional) fine-tuning to further refine the models' responses to the specific language of DPAs.

Subsequently, the generated responses are analyzed to determine the compliance of the DPA text with the GDPR, identifying areas of compliance and non-compliance in a comprehensive report (Fig. 4.1, Step ❸). An example of the model's explanation and justification is provided in Fig. 4.3, for the input text without context (❶), and for the case where the paragraph-level context is provided (❷).

4.5 Implementation

Our approach, described in Section 4.4, has been implemented in Python. We publicly release our implementation [100] to encourage future research and facilitate replication. The

state-of-the-art LLMs that we experiment with are: *Phi-2* [122], *Mistral-7B* [119], *Mistral-7B-Instruct* [120], *Zephyr-7B* [123], *Mixtral-8x7B-Instruct-v0.1* [121], all from Hugging Face [116], using the Transformers 4.35.2 library operated in PyTorch 1.10.2+cu113, as well as *gpt-3.5-turbo-0125* and *gpt-4-0125-preview* from OpenAI [173] through the OpenAI Python package. We further implement one baseline approach for comparison: *bert-base-cased* from Hugging Face [116]. To quantize and further fine-tune the LLMs, we use the Transformers 4.35.2 library, Peft library version 1.5.3, and the trl library version 0.7.10.

4.6 Plan for the Evaluation

Our proposed approach is designed to be applicable across a variety of legal documents. Here, we instantiate it for DPAs, focusing on determining the effectiveness of LLMs in identifying GDPR compliance. DPAs are chosen for their complex and detailed nature, providing a rich basis for evaluation within the RE community [10, 11].

4.6.1 Research Questions

RQ1: How do state-of-the-art LLMs, both open-source and closed-source, fare against one another in terms of accuracy for zero-shot learning and fine-tuning in regulatory compliance tasks? This research question compares generative LLMs in terms of their ability to understand and apply regulatory requirements. To answer RQ1, we use *Accuracy*, *Precision*, *Recall*, and *F-score* metrics.

RQ2: Compared to traditional sentence-level analysis, how does incorporating paragraph context and the textual specification of compliance rules enhance the performance of compliance checking? This research question aims to assess the enhancement in accuracy of compliance checking brought about by the integration of paragraph-level context and rules, as opposed to traditional single-input sentence-level methods. To answer RQ2, we report the performance using *Accuracy*, *Precision*, *Recall*, and *F-score* metrics.

RQ3: What are the cost and time implications associated with our proposed approach? RQ3 provides a practical assessment of the approach's resource efficiency, considering time and costs in terms of computing and financial resources.

4.6.2 Dataset

Our evaluation uses a dataset of DPAs, borrowed from Amaral et al. [10], which has been manually annotated with GDPR compliance decisions. This dataset, which we subjected to further manual preprocessing to ensure data quality, has been built at the sentence level. This dataset is partitioned into training (P) and evaluation (E) sets. It comprises 46 compliance rules derived from GDPR Article 28, covering crucial aspects such as identification of controllers and processors, duration and nature of data processing, breach reporting

requirements, and data-subject rights. These compliance rules serve as classification labels for assessing whether specific DPA passages satisfy GDPR requirements. The top 25 most frequently occurring rules are listed in Appendix B of this thesis.

4.6.3 Experiments

This section describes the experimental setup and procedures designed to answer our research questions.

Experiment I. This experiment answers RQ1. Following Step ❶ of our approach (Section 4.4), for every document $p \in P$ from [10], we do data preparation through content chunking. The input DPAs are segmented into sentences and paragraphs (passages) to incorporate paragraph-level data.

Next, prompt templates, compliance rules, and data chunks are paired to construct the prompt as per Step ❷ of our approach, with prompt details further elaborated in Fig. 4.2. The selected LLMs are configured with the temperature hyperparameter set to 0.2. This minimizes variation in responses, thereby making outputs more deterministic while maintaining the ability to generate diverse responses. For each LLM, we conduct zero-shot experimentation on both sentence-level and paragraph-level passages (Step ❸ of our approach). The performance is measured against a gold standard set [10] using the evaluation metrics listed in Section 4.6.4.

Optionally, fine-tuning is conducted on the best-performing models from the zero-shot experiments. Fine-tuning is likely to better guide the models in following instructions by exposing them to a set of examples. All reported results in this chapter are from zero-shot prompting without fine-tuning.

Experiment II. This experiment answers RQ2. We compare the best configuration of our approach resulting from EXPI with single-input models like BERT [52], which process sentences in isolation, lacking broader context and explicit rule definitions; see Section 4.6.4 for the evaluation metrics used.

Experiment III. This experiment answers RQ3 by measuring the execution time and cost of our approach from the perspective of end-users to ensure its feasibility for real-world applications. We use a modest platform to replicate the resources available to end-users. Specifically, we use the free version of Google Colab Cloud with the following specifications: Intel Xeon CPU@2.30GHz, Tesla T4 GPU, and 13GB RAM. To conduct the GPT experiments, we utilize OpenAI's token-based plan (paid subscription service).

4.6.4 Metrics

For each input passage (sentence of paragraph) m from E (as defined in Section 4.6.2), we have the model predict whether m satisfies each compliance rule R. We evaluate the quality of the predictions using *Precision*, *Recall*, *F-score*, and *Accuracy*, according to their standard definitions.

A true positive (TP) arises when the model correctly predicts R_j as "satisfied", and a true negative (TN) when it correctly predicts R_j as "not satisfied". A false positive (FP) arises when R_j is incorrectly predicted as "satisfied", while a false negative (FN) arises when R_j is incorrectly predicted as "not satisfied".

4.7 Results

4.7.1 Answers to Research Questions

Below, we provide initial answers to our research questions, drawing from our ongoing experimentation and following the analysis procedures detailed in Section 4.6.

RQ1. Our experiments with *Phi-2* [122], *Mistral-7B* [119], *Mistral-7B-Instruct* [120], and *Zephyr-7B* [123] indicate that these LLMs fare poorly as alternatives for our intended purposes. There are multiple cases where these LLMs do not follow predefined instructions in the prompt. Even when fine-tuned with sentence-level passages, these models did not exhibit the desired behaviour. We also note that, in view of Jiang et al.'s findings [132], indicating that *Mistral-7B* [119] outperforms both *LLaMA2-7B* [118] and *LLaMA-13B* [117] on several benchmarks, we did not experiment with these models.

In addition, our results suggest that Mixtral-8x7B-Instruct-v0.1 [121] and GPT models outperform previous LLMs in understanding and following instructions as specified in the prompts in zero-shot learning scenarios. Indeed, our experiments with *gpt-3.5-turbo-0125* [173], *Mixtral-8x7B-Instruct-v0.1* [121], and *gpt-4-0125-preview* [173] show promising improvements when transitioning from sentence-level to paragraph-level passages. Table 4.1 shows the improvements in the mean *Accuracy* results obtained using these three models.

Table 4.1: Mean accuracy results

Model	**Initial Accuracy (%)** (Sentence Level)	**Improved Accuracy (%)** (Paragraph Level)
GPT-3.5-Turbo-0125	30	63
Mixtral-8x7B-Instruct-v0.1	33	69
GPT-4-0125-Preview	41	81

Time constraints limited the scope of our evaluation. Nonetheless, initial findings through our zero-shot experiments support the superior analytical capabilities of these LLMs when broader textual contexts are integrated.

RQ2. The BERT-based classification approach by Ilyas et al. [127] omits several compliance rules in its implementation and furthermore lacks annotated paragraph-level data.

These issues preclude, at this stage, a conclusive comparison of our approach against that of Ilyas et al.'s to answer RQ2. To conduct a tentative comparison, considering that Ilyas et al. do not provide a public implementation of their approach, we have re-implemented it. This re-implementation, publicly available in our online repository [99], serves as our benchmark for comparison. The results obtained from this benchmark across two compliance rules with high prevalence yield an average *F-score* of 67%. In contrast, the lowest *F-score* observed in our approach (using *gpt-4-0125-preview* as the underlying LLM) exceeds 80%, indicating major accuracy gains. Further analysis details can be found in our online material [100].

RQ3. In our zero-shot learning experiment across a typical DPA, the total token count, total assistant token count, and the combined token count of both user and system are 25473, 769, and 24479, respectively. At the time of the experimentation (January 2024), the cost of processing our example DPA with GPT-3.5 Turbo and with GPT-4 Turbo using the OpenAI API was $0.026 and $0.27, respectively, subject to the OpenAI pricing at that time. The costs for an individual DPA are minimal (noting, of course, the order-of-magnitude disparity between GPT-3.5 Turbo and GPT-4 Turbo costs), and are expected to generate significant savings for lawyers, compliance experts, and consequently, businesses seeking legal services. In interviews, our collaborating industry partner estimated that they dedicate two weeks and a team of three experts to each business. They foresee automation cutting mundane tasks and compliance costs, while emphasizing the indispensable role of human expertise alongside automation. Lastly, in relation to execution time, we observe that our approach has an average execution time of approximately 0.7 seconds per paragraph. This suggests our implementation can handle more than 5000 paragraphs of textual legal content per hour. Considering that our approach can be run offline, we find the performance of our approach to be acceptable.

4.7.2 Limitations and Validity Considerations

Generalizability: Our evaluation is preliminary. While the approach may work well for the specific datasets and compliance rules tested, we do not have enough evidence to conclude that it would generalize to other domains of law or regulatory frameworks. Furthermore, we observed that smaller models, even the instruction-tuned models (e.g., Mistral-7B [119], Zephyr-7B [123]), often failed to follow prompts correctly, even with fine-tuning. This indicates that model size and architecture play a crucial role in ensuring consistent instruction adherence, which may affect the approach's generalizability across different LLMs. Also, as our results report *Accuracy*, we acknowledge this limitation and plan to include the full set of metrics in future work once the test set has been fully annotated. **Interpretability of Justifications:** While generative LLMs may provide justifications for their decisions, the interpretability of these justifications is crucial. They need to be clear and understandable to legal experts to be useful. Our evaluation has not yet examined the usefulness of the automated justifications provided. **Context Span:** While adding context that spans across sentences is beneficial, the context might also span multiple paragraphs, which could be considered as additional input for each prompt. Recent advancements in LLMs, e.g., the development of Gemini 1.5 Pro [78] with a context

window of up to 1 million tokens, facilitate such extended context. However, the challenge of selecting the appropriate amount of context is still critical to avoid the "needle in a haystack" problem, where too much information can dilute and obscure relevant details. Our current evaluation suggests that paragraphs are a better context than sentences for compliance checking; however, the evaluation does not address the question of what the "optimal" context is for this task.

Evolution: Most legal texts are revised continuously over time. Continuously updating the training data for LLMs is essential to ensure accuracy and relevance, distinguishing between up-to-date information and outdated "hallucinations" not aligned with current regulations. Our current work does not address the potential risks posed by deprecated legal texts and the difficulty of handling multiple versions of the same legal text. **Content Chunking:** While investigating DPAs, we discovered through experiments that the optimal chunk of content, containing reasonable and necessary context for a sentence, is extracted when a line break followed by a heading is observed. Our current evaluation does not explore other regulatory artifacts, e.g., privacy policies, to determine the most suitable chunk for each sentence based on its context. **Model Bias:** LLMs may inherit biases from their training data, which can affect the fairness and impartiality of compliance assessment. Our current evaluation does not offer insights into the perceived and actual severity of this risk.

4.8 Data Availability

Our online repository is available at [100]. Specifically, the dataset can be found at [98], while the implementations of algorithms and evaluation scripts are provided at [99].

4.9 Future Research

The research presented in this chapter aims to make a case for the need to reconsider current practices in legal compliance automation in light of recent advances in AI. Specifically, we argue that the substantially enhanced capacity of modern LLMs to handle context is likely to induce a major shift in our treatment of textual legal artifacts. This shift will involve transitioning from analyzing smaller contexts, such as individual sentences and phrases, to considering larger volumes of content, such as paragraphs and beyond, as context. We posit that the larger context will be able to provide the prerequisite knowledge, including cross-referenced legal materials, to create a self-contained basis for accurate automated decision-making regarding compliance and non-compliance.

Our future work will focus on four main aspects: (1) enriching the DPA dataset [10] with paragraph-level annotations; (2) conducting comprehensive empirical evaluations to validate the effectiveness of paragraph-level context in increasing LLM accuracy, as defined in Section 4.6; (3) benchmarking against prior BERT-based approaches, e.g., [127], to showcase comparative advantages; and (4) seeking input from legal experts to review the outputs, particularly focusing on the explanation and justification provided by LLMs.

Chapter 5

From Law to Gherkin: A Human-Centred Quasi-Experiment on the Quality of LLM-Generated Behavioural Specifications from Food-Safety Regulations

Context: Laws and regulations increasingly impact software design, development, and quality assurance in many regulated domains. Yet, legal provisions are typically expressed in technology-neutral language to maintain broad applicability and long-term relevance. This poses challenges for software engineers, who must develop artifacts such as specifications, requirements, and acceptance criteria to determine whether software and systems are compliant with relevant regulations. Creating software engineering artifacts manually for legal compliance is labour-intensive, error-prone, and requires extensive domain expertise. Recent advances in Generative AI (GenAI), particularly LLMs, offer a promising avenue for exploring the automated derivation of software engineering artifacts from legal texts.

Objective: We present the first systematic human-subject evaluation study of LLMs' capability to automatically derive behavioural specifications from legal texts through a quasi-experimental design. These behavioural specifications constitute automatically derived software artifacts that capture legal requirements, making legal texts more palatable to a wider range of stakeholders, including software engineers.

Methods: We engage 10 participants for our quasi-experimental study who evaluate the quality of behavioural specifications generated from legal text by two leading LLMs, Claude and Llama. We ground our study in food-safety regulations, selected for their broad applicability and leveraging our existing domain knowledge. We employ Gherkin – a structured language widely used in agile software development and BDD – as our target specification format. Using Claude and Llama, a total of 60 Gherkin specifications are generated from food-safety regulations. Each of the 10 participants independently assesses a subset of specifications (12 each) across five quality criteria: *Relevance*, *Clarity*, *Completeness*, *Sin-*

gularity, and *Time Savings*. Each specification is evaluated by two participants, resulting in a total of 120 assessments. Participants rate each criterion using predefined scales and provide qualitative feedback.

Results: Across 120 assessments, for *Relevance*, 75% of responses fall into the highest rating category, and an additional 20% in the second-highest category, with no ratings indicating complete irrelevance. *Clarity* is rated in the highest category in 90% of cases and in the second-highest category in 10%. For *Completeness*, 75% of ratings are in the highest category, with an additional 19.2% in the second-highest one. For *Singularity*, 81.7% of responses received the highest rating category, with 11.7% in the second-highest category and 6.7% indicating more substantial blending of multiple intents. Finally, for *Time Savings*, 67.5% of responses indicate the highest rating category in terms of reducing manual effort, while 24.2% are in the second-highest category, with no responses rated the lowest category.

Claude and Llama – the two LLMs examined – both achieve the highest rating category for *Relevance* in 75% of cases. Llama slightly outperforms Claude in *Clarity* (91.7% vs. 88.3% highest rating category) and *Completeness* (76.7% vs. 73.3%). Claude shows slightly better performance in *Singularity* (85% vs. 78.3%), whereas Llama performs better in *Time Savings* (71.7% vs. 63.3%). No statistically reliable differences were observed across participants or between LLMs. Qualitative feedback points to some issues, including hallucinations, omissions, and mixed-intent scenarios, while also offering approvals that reinforce the utility of the generated Gherkin specifications.

Conclusion: LLMs show strong performance in automating the generation of high-quality Gherkin behavioural specifications from food-safety regulations. This automation can reduce manual effort by offering legal provisions into a structured, developer-friendly format, usable for software implementation, quality assurance, and facilitating the generation of executable test cases.

5.1 Introduction

Laws and regulations increasingly influence software engineering practices, shaping the design, development, and assurance processes across regulated domains. For instance, in the financial sector, compliance with standards such as the Payment Card Industry Data Security Standard (PCI DSS) [180] ensures the secure handling of sensitive cardholder information. In healthcare, regulations such as the Health Insurance Portability and Accountability Act (HIPAA) [217] govern the secure management of confidential patient information. These regulatory frameworks define functional requirements as well as legal, safety, and security mandates that could complicate software development and quality assurance. Failure to comply can result in serious consequences, e.g., financial penalties, reputational damage, and risks to public safety [137, 141, 177]. In this context, misinterpretation or oversight in translating laws and regulations into software artifacts (such as behavioural specifications) can become very costly.

Legal texts are commonly phrased in technology-agnostic language to maintain broad

applicability and long-term relevance. This can, nonetheless, complicate their application in concrete technological contexts [46, 107]. Translating legal texts into software engineering artifacts, such as behavioural specifications or acceptance criteria, remains a significant challenge. Practitioners often have difficulty interpreting legal texts in a way that leads to precise, verifiable requirements and acceptance criteria necessary to demonstrate compliance. Manual translation is time-consuming, error-prone, and inconsistent, increasing the risk of non-compliance, especially in high-stakes domains where regulatory adherence is critical to public safety [15, 56, 142]. Existing research in Requirements Engineering (RE) has primarily tackled legal compliance through methods of requirements extraction, classification, and ambiguity resolution from legal provisions [2, 12, 36, 43, 51, 107, 112, 158, 213, 239, 240]. Yet, little attention has been given to systematically transforming complex legal texts into structured, executable software artifacts such as behavioural specifications, an important angle that our research aims to address.

5.1.1 Motivation

Recent advancements in Generative AI (GenAI), particularly LLMs, present a promising opportunity to provide executable "Given—When—Then" behavioural specifications from legal provisions, which are typically written in dense legalese, thereby bypassing traditional multistage, manual processes. Early efforts have shown that NLP techniques can be used to generate software engineering artifacts – such as acceptance tests, test cases, code stubs, and control flow graphs (test paths) – from stakeholder-written specifications, natural-language requirements, and UML use-case descriptions [68, 143, 204]. These approaches mostly depend on rule-based and heuristic NLP methods, graph traversal, and pattern matching. More recently, researchers have begun exploring the use of LLMs for generating software artifacts, particularly test cases. Mathur et al. [159] employ prompting to derive unit tests from requirement statements. Ferreira et al. [66] generate Gherkin scenarios and test cases from user stories in a web-application setting. Alagarsamy et al. [8] fine-tune LLMs to produce syntactically correct Java unit tests directly from method-level descriptions. Although these studies advance automated test generation in software engineering contexts, they do not address the application of LLMs to legal texts.

In this chapter, we present the first human-subject study that systematically assesses the effectiveness of two state-of-the-art LLMs – Llama and Claude – in generating behavioural specifications from legal provisions. Our study is framed using Basili's Goal-Question-Metric (GQM) paradigm [26]. The overarching goal is to evaluate how well Llama and Claude can perform in getting the artifact derivation done, i.e., generate Gherkin specifications from legal provisions, with the aim of making legal texts more accessible to software engineers. The concrete research questions and evaluation metrics used to operationalize this goal are detailed in Section 5.1.3. We employ Gherkin, a widely adopted language within agile and Behaviour-Driven Development (BDD) methodologies [55, 193, 226], to represent extracted behavioural specifications. Gherkin's syntax makes it possible for stakeholders from diverse backgrounds to understand, validate, and actively participate in the RE process, thus improving collaboration and transparency [193].

We address a specific instance of the broader challenge of translating legal texts into software artifacts, focusing on the domain of ***food safety***. We choose this domain for three main reasons: First, food safety is a high-impact domain affecting all individuals [234] and representing a substantial global market [189]. Second, food safety is increasingly intertwined with Industry 4.0 and IoT-based monitoring systems, leading to software-relevant legal requirements; however, because food-safety regulations are written in technology-neutral language, a gap remains between legal provisions and the software systems required to implement them [33, 47]. Third, food safety offers the advantage of allowing us to build on our prior experience and a previously developed conceptual model of software-relevant concepts in this domain [107], supporting deeper reflection on the challenges and the formulation of precise guidelines for our study.

5.1.2 Illustrative Example

Figure 5.1 shows an example Gherkin specification, based on a legal provision incorporated by reference into the SFCR [83][1], and generated using Claude 3.7 Sonnet [17] and Llama 3.3 70B Instruct [124]. The provision is as follows:

> Liquid Whole Egg, Dried Whole Egg and Frozen Whole Egg are the foods that meet the standard set out in section B.22.034 of the FDR and
>
> (a) in the case of liquid whole egg and frozen whole egg, contain at least 23.5% egg solids by weight;
>
> (b) in the case of dried whole egg, contain not more than 5% water; and
>
> (c) contain not more than
>
> (i) 50,000 viable bacteria per gram; and
>
> (ii) 10 coliform organisms per gram.

Different LLMs may generate Gherkin specifications that vary in quality across different criteria. For instance, the Gherkin specification shown in Fig. 5.1(a) is generated by Claude 3.7 Sonnet [17]. It follows BDD best practices by presenting scenarios that each focus on a single, well-defined objective. In contrast, the specification in Fig. 5.1(b), generated by Llama 3.3 70B Instruct [124], blends multiple purposes within individual scenarios, such as 1) measuring product weight, 2) assessing water content, and 3) evaluating microorganism levels. BDD scenarios are generally more effective when each scenario addresses only one distinct goal. Our objective in this is to evaluate, through a quasi-experimental study, the quality of Gherkin specifications generated by LLMs from legal texts, and to identify any recurring issues in such specifications.

[1] https://bit.ly/4o4tEDq

```gherkin
Feature: Egg Product Requirements
  As a food safety regulator
  I want to ensure that egg products meet the standard set out in
  section B.22.034 of the FDR
  So that I can verify their compliance

  Background:
  Given an egg product ready for analysis

  Rule: Egg products must meet the FDR standards for composition
  and bacteria limits

  Scenario: Verify composition for Liquid and Frozen Whole Eggs
  Given the product type is liquid whole egg or frozen whole egg
  When the product is analyzed
  Then it must contain at least 23.5% egg solids by weight

  Scenario: Verify composition for Dried Whole Eggs
  Given the product type is dried whole egg
  When the product is analyzed
  Then it must contain not more than 5% water

  Scenario: Verify bacteria and coliform limits for all egg
  products
  When the product is analyzed for bacteria content
  Then it must not contain more than 50,000 viable bacteria per
  gram
  And it must not contain more than 10 coliform organisms per gram
```

(a) Generated by Claude 3.7 Sonnet: each scenario addresses a distinct requirement for different egg types.

```gherkin
Feature: Egg Product Requirements

  Scenario: Liquid Whole Egg requirements
  Given a liquid whole egg product
  When the product is analyzed
  Then it must contain at least 23.5% egg solids by weight
  And it must not contain more than 50,000 viable bacteria per
  gram
  And it must not contain more than 10 coliform organisms per
  gram

  Scenario: Frozen Whole Egg requirements
  Given a frozen whole egg product
  When the product is analyzed
  Then it must contain at least 23.5% egg solids by weight
  And it must not contain more than 50,000 viable bacteria per
  gram
  And it must not contain more than 10 coliform organisms per
  gram

  Scenario: Dried Whole Egg requirements
  Given a dried whole egg product
  When the product is analyzed
  Then it must contain not more than 5% water
  And it must not contain more than 50,000 viable bacteria per
  gram
  And it must not contain more than 10 coliform organisms per gram
```

(b) Generated by Llama 3.3 70B Instruct: each scenario combines composition and microbiological checks, violating *Singularity* and reducing *Clarity* of individual objectives.

Figure 5.1: Example Gherkin specifications generated by two different LLMs for the same food-safety legal provision: (a) illustrates singular and focused scenarios; (b) illustrates mixed-objective scenarios.

5.1.3 Research Questions (RQs)

To operationalize our GQM goal (Section 5.1.1), we answer the following RQs:

RQ1: *How well do LLMs perform at generating Gherkin specifications from food-safety legal provisions?*

RQ2: *How do Llama and Claude compare when generating Gherkin specifications from food-safety legal provisions?*

We answer RQ1 and RQ2 by evaluating the quality of LLM-generated Gherkin specifications along five criteria: *Relevance*, the degree to which the specification matches the intended system behaviour as described in the legal text; *Completeness*, the degree to which all functions and characteristics implied by the legal requirement are included without omissions; *Clarity*, the degree to which the specification is written in a clear and unambiguous manner, allowing unique interpretation; *Singularity*, the degree to which scenarios focus on one purpose, avoiding mixed-purpose scenarios. A singular scenario addresses a distinct, atomic objective. This contrasts with "mixed-purpose" scenarios, which conflate multiple independent requirements (e.g., verifying both physical composition and microbial limits) into a single test flow; and, *Time Savings*, the degree to which the generated specification can be reused or adapted, thus reducing manual authoring effort. We note that, in our study, *Time Savings* is based on a self-reported judgment of efficiency, not on direct time-tracking. Each criterion is rated on a five-point scale. Definitions and the scale's full description appear in Table 5.1 (Section 5.3.3).

5.1.4 Contributions

We present a process that (1) generates Gherkin specifications using LLMs, and (2) evaluates the quality of the generated specifications through a quasi-experiment [131] with human participants. Our main contributions are:

1. **LLM-based derivation of Gherkin specifications from legal texts:** We introduce a process that employs LLMs to transform legal provisions into structured Gherkin specifications.
2. **Empirical evaluation with human participants:** We conduct a quasi-experimental study to systematically evaluate the quality of LLM-generated specifications in terms of *Clarity*, *Completeness*, *Relevance*, *Singularity*, and *Time savings* using participant ratings. We further analyze qualitative feedback provided by participants to identify recurring issues and areas for improvement in the generated specifications.

5.1.5 Structure

Section 5.2 provides background. Section 5.3 details (i) our LLM-based approach for deriving Gherkin specifications from legal texts and (ii) our quasi-experimental study design. Section 5.4 reports on the results of our study, providing a descriptive analysis of the generated Gherkin specifications as well as the quantitative results that answer RQ1 and RQ2. Section 5.5 outlines qualitative insights from the study. Section 5.6 discusses risks and implications of our findings. Section 5.7 addresses threats to validity. Section 5.8 compares our work with related research. Section 5.10 provides concluding remarks. In the spirit of open science, Section 5.9 gives an overview of our replication package.

5.2 Background

This section presents the necessary background on food-safety regulations and the enabling technologies used.

5.2.1 Food-Safety Regulations

Countries worldwide maintain regulatory frameworks to ensure a safe food-supply chain [234]. In Canada, the Safe Food for Canadians Regulations (SFCR) [83] consolidate multiple federal regulations relevant to importing, exporting, and inter-provincial trading of food products. SFCR covers both general and product-specific requirements; for the latter, it defers detailed obligations to the Food-Specific Requirements and Guidance (FSRG) regulations [81], which address domains like dairy, eggs, fish, produce, meat, honey, and maple. We use SFCR and FSRG texts in our evaluation. Like all legal texts, SFCR and FSRG contain cross-references intertwining their content with other legal documents, including their enabling act, the SFCA, and related regulations such as the FDR, enabled by the Food and Drugs Act (FDA). The scope of our investigation into FDR, FDA, and SFCA is driven by explicit cross-references identified within the evaluated SFCR and FSRG texts.

This study builds on prior work by Hassani et al. [107] that developed a taxonomy of software-relevant concepts in food-safety regulations. That study classified software-relevant provisions from Canadian and U.S. food-safety regulations and released a dataset of software-related provisions, labelled for their relevance to software based on the proposed taxonomy. A subset of these provisions (30 provisions) [105] serve as the input for our study in this current .

5.2.2 Behavior-Driven Development (BDD)

Behaviour-Driven Development is a collaborative software engineering method designed to express requirements in a human-readable format, making them accessible to both technical and non-technical stakeholders [179]. BDD relies on concrete scenarios, following a Given–When–Then structure, to describe system behaviours based on user needs or

domain knowledge. In BDD, each scenario should focus on a single objective. This approach improves design, development, and testing clarity by reducing ambiguity, minimizing miscommunication among stakeholders, and simplifying verification or compliance checking [179]. Avoiding combined or overlapping scenarios makes it straightforward to trace failures directly linked to one specific behaviour. BDD has seen increasing adoption in practice. Yet, a systematic mapping study by Binamungu et al. [31], covering 166 publications (2006–2021), highlights open challenges, including limited research on scenario quality metrics [31]. Binamungu et al. [31] propose suite-level quality principles such as vocabulary consistency, abstraction management, and redundancy minimization which complement scenario-level metrics. Our study uses established scenario-level metrics (discussed in Section 5.8.2) to evaluate specifications generated from legal texts using LLMs.

In this chapter, we adopt the Gherkin syntax, a scenario-based language widely used within BDD methodologies, to specify, in a structured format, the functions and characteristics envisaged by legal texts.

5.2.3 Gherkin Language

The Gherkin language is central to BDD, providing a structured Given–When–Then template that is both human- and machine-readable. Gherkin's syntax facilitates automation by enabling specifications to be directly converted into executable test cases given appropriate step definitions. Popular tools such as JBehave [130] and Cucumber [111] use Gherkin to enable seamless integration between behaviour specifications and automated testing. Cucumber, in particular, is a testing framework that matches Gherkin feature files with testing code, allowing developers to run code that automates the scenarios [111, 156]. While our study does not implement test automation, it lays the groundwork for such integration in the context of legal compliance.

Gherkin specifications are organized using a small set of keywords: `Feature`, `Background`, `Scenario`, `Given`, `When`, `Then`, `And`, `But`, `Scenario Outline`, and `Examples` [156]. Figure 5.1 (in Section 5.1) shows example Gherkin specifications that include most of these keywords. A `Feature` describes a coherent piece of functionality or requirement, typically grouping multiple related scenarios. Each `Scenario` follows the Given–When–Then pattern: `Given` defines the context, `When` describes the event or action, and `Then` asserts the expected outcome [156]. Gherkin also supports additional syntax elements such as `"""` (doc strings), `|` (data tables), `@` (tags), and `#` (comments) [111].

Our study focuses on Gherkin specifications. We observe that LLM-generated Gherkin specifications occasionally include user stories [24, 150] at the feature level. A user story is another common BDD artifact used to capture requirements, usually using the pattern: *As a <Role>, I want <Functionality>, so that <Goal>*. These stories typically help clarify the broader legal context, the intent behind the feature, and the value provided.

5.2.4 Large Language Models (LLMs)

LLMs have already proven useful in many requirements engineering tasks such as elicitation and refinement [22], motivating our attempt to apply them for deriving software requirements from legal texts. LLMs are statistical models trained on extensive text corpora. Through training, LLMs develop a nuanced understanding of language patterns and construct sophisticated syntactic and semantic representations. This, in turn, enables LLMs to generate coherent and contextually appropriate text across a wide range of domains, making them valuable for tasks involving complex language interpretation and generation.

Models such as Llama and Claude are representative examples of modern LLMs. Our study compares the capabilities of these two models in translating food-safety regulations into Gherkin specifications. A brief overview of these two LLMs is provided below:

Large Language Model Meta AI (Llama) 3.3 70B Instruct, developed by Meta [124, 216], is a 70-billion-parameter, instruction-tuned LLM optimized for multilingual dialogue and complex reasoning tasks. Notably, Llama 3.3 70B Instruct demonstrates enhanced capabilities in reasoning, mathematics, and instruction-following, achieving performance levels comparable to the larger Llama 3.1 405B model while requiring only a fraction of the computational resources. Its instruction-tuned architecture enables it to outperform many existing open-source and closed-source chat models on industry benchmarks [14, 53, 144].

Claude 3.7-Sonnet, developed by Anthropic, was the latest version of the Claude model series at the time of our study. This closed-weight LLM offers strong capabilities in natural-language understanding and generation [17]. It introduces a novel "hybrid reasoning" approach, enabling both quick answers and detailed, step-by-step reasoning based on task requirements. This flexibility allows it to handle complex language patterns while maintaining structural and stylistic accuracy, making it well-suited for areas like creative writing. Its "extended thinking" mode further improves its performance in fields that require careful reasoning, such as mathematics and scientific problem-solving.

5.3 Methodology

This section presents the research process we adopt in order to investigate the effectiveness of using LLMs to derive Gherkin specifications from legal provisions.

Our study consists of three main phases: (1) deriving Gherkin specifications from legal provisions using LLMs, (2) evaluating the resulting specifications through participant ratings and feedback, and (3) analyzing the collected responses both quantitatively and qualitatively. This process involves recruiting participants, generating specifications using LLMs, conducting human evaluations, and synthesizing input from participants.[2]

For our study design, we concluded that a fully controlled experiment was infeasible under our real-world constraints, as it would require rigid standardization of numerous

[2]Ethics approval for our study was granted by the University of Ottawa's Research Ethics Board (file number H-08-24-10663).

potential confounders – time of day, participant fatigue, and ambient environmental conditions – that we simply could not enforce given our participants' varied schedules. Moreover, we deliberately recruited participants whose experience ranged from senior undergraduates to final-year PhD students in order to preserve external validity; narrowing this spectrum would have both limited recruitment and reduced the relevance of our findings. Taken together, the logistical challenge of strict control over extraneous factors and the need to accommodate a diverse cohort would have significantly undermined the internal validity of a fully controlled experiment.

Similarly, a randomized crossover design [162], where each participant is exposed to all treatments in a random order, was ruled out since we wanted (1) every Gherkin specification to be reviewed by two participants, and (2) each participant to review six specifications generated by Llama and six by Claude. These constraints made it infeasible to randomly assign treatments while ensuring balanced task coverage. Therefore, our treatment assignment was driven by study-design constraints and participant-task balancing, rather than full randomization.

Because we cannot control all key experimental variables, our study employs a *quasi-experimental* design. As described by Jedlitschka et al. [131], a quasi-experiment is an empirical inquiry similar to a controlled experiment, except that treatment assignment is determined by subject or object characteristics rather than full random allocation. In our case, while task selection for each participant involves random sampling to ensure task coverage without overlap, the overall assignment is guided by study-design requirements, specifically balanced exposure to both models (Claude and Llama), while also ensuring that no participant evaluated Gherkin specifications for the same legal provision from both models.

5.3.1 Participant Recruitment

The first step is participant selection. Eligibility is determined by the following:

1) *Technical Background*: A) Proficiency with BDD principles and Gherkin syntax. B) Experience in programming, writing and reviewing test cases. C) Completion of specific university-level courses such as "Software Requirement Analysis" and "Software Quality Assurance" ensuring foundational knowledge of requirements and test specification; 2) *Language Proficiency*: Ability to comprehend and produce written work in English, as study materials and instructions are provided in English; and 3) *Open Eligibility*: In alignment with Equity, Diversity, and Inclusion (EDI) principles, no restrictions were applied based on race, sex, or geographic location. However, since we wanted to compensate participants, we focused on the pool of students at the University of Ottawa.

To verify eligibility, we asked interested participants to send us a short email describing their background. Specifically, they were instructed to: 1) indicate whether they had completed the relevant courses and when; 2) describe their familiarity with BDD and Gherkin syntax; 3) summarize their experience with writing test cases and reviewing software requirements. Participants were also asked to attach a copy of their CV to support

their self-reported qualifications. We reviewed this information to ensure that all selected participants met the minimum technical criteria.

Participants were recruited through multiple channels in the Software Engineering and Computer Science programs at the University of Ottawa. The recruitment strategies included: 1) *In-Class Presentations*: Short presentations in advanced Software Engineering courses explained the study scope and voluntary nature of participation. 2) *Post-Class Flyers*: Flyers were distributed with detailed study information and instructions for potential volunteers on how to sign up [105]; and 3) *Emails and Announcements*: Targeted emails were sent to eligible students in relevant courses, detailing the tasks and compensation structure.

Participants were included on a first-come, first-served basis, subject to eligibility, to ensure fairness. When more candidates responded than needed (target sample size: 10), those not selected received follow-up emails thanking them for their interest and explaining that recruitment had closed.

Participants were compensated for their time and effort. The compensation structure was as follows: 1) *Per Task*: $8 for each completed Gherkin evaluation task. 2) *Completion Bonus*: An additional $54 for completing all 12 tasks. 3) *Total Compensation*: Participants could earn up to $150. This tiered structure ensured participants could withdraw at any time while still receiving proportional compensation for completed tasks.

All participants were required to review and sign a consent form [105], detailing the study goals, procedures, and the right to withdraw. The consent process ensured that participation was informed, voluntary, and conducted with respect to ethical standards, including anonymity. Data collection for the study was conducted between February 2025 and March 2025.

5.3.2 LLM-based Gherkin Specification Generation

In the second step, we have an LLM (Llama or Claude) generate Gherkin specifications from legal provisions. This step was carried out via the respective chat interfaces for these models, namely the Hugging Face chat interface for Llama and the Claude web interface. As these interfaces do not expose model parameters to the user, all generations were performed using the default settings.

Figure 5.2 shows our prompt template. To improve the consistency, task alignment, and interpretability of LLM outputs, our prompt design follows established prompt-engineering principles: (i) *clear and specific task definition*, (ii) *role assignment to guide the model's perspective* [140, 235], and (iii) *contextual grounding through domain-specific information* [18, 176, 178].

The prompt has three main components: a glossary of domain-specific concepts, a food-safety legal provision, and a task instruction. The glossary is drawn from our prior work [107], which defines concepts such as *Data*, *Measurement*, and *Time Constraint* to support consistent interpretation of software-relevant food-safety provisions. The provision input is sampled from the data released by Hassani et al. [107]. The task instruction asks

Figure 5.2: Prompt for generating a Gherkin specification.

```
System:
You are a senior Requirements Engineering expert. Your task is to
transform legal provisions into Gherkin specifications. Always
consider the concepts and definitions provided in the glossary
for food safety legal provisions.

Glossary:
Data: any information used to convey knowledge, provide assurance
or perform analysis;
Label Data: information that a food-product package or container
must bear;
Non-label Data: any food-safety-relevant data other than label
data that needs to be collected and/or retained for inclusion in
documents such as certificates, reports, guarantees and letters.

Measurement: Association of numbers with physical quantities;
    Colour: (self-evident);
    Firmness: degree of resistance to deformation;
    Mass: amount of substance by weight or volume;
    Pathogen: a microorganism that causes disease;
    Size: dimension (e.g., length or thickness) or surface area;
    Temperature: (self-evident);
    Water Content: humidity or moisture.

Time Constraint:  A temporal restriction, which in our context,
is expressed using intervals, deadlines or periodicity.

User:
Here is a software-relevant legal provision: {{Legal Provision}}.
Transform the legal provision into a Gherkin specification that:
- Uses Given-When-Then steps.
- Reflects the glossary concepts where applicable.

Assistant:
Answer.
```

the LLM to produce a behavioural specification in the Gherkin syntax, aligning each provision with relevant glossary concepts and structuring the output in the `Given-When-Then` format.

5.3.3 Participant Evaluation of the Resulting Gherkin Specifications

In the third step, participants evaluated each generated Gherkin specification against the defined quality criteria. Before performing their tasks, participants received:

Training Materials: 1) A brief tutorial on BDD principles and Gherkin syntax, clarifying how Gherkin can be used for legal compliance; and 2) Example Gherkin specifications that illustrate pitfalls and best practices (e.g., incomplete scenarios, and single purpose per scenario) [105].

Evaluation Instruction: 1) A concise explanation of each quality criterion – namely *Relevance*, *Completeness*, *Clarity*, *Singularity*, and *Time Savings* – with ordinal scale definitions to ensure consistent scoring (Table 5.1), together with instructions to provide brief qualitative feedback when problems with the specification are identified. 2) Instructions to conduct a plausibility check, defined as a binary assessment of whether a specification could be realized in a real-world implementation. A specification is considered plausible if it could reasonably occur in practice without violating fundamental physical laws (e.g., conservation of mass) or containing logically inconsistent conditions, and if it aligns with common-sense expectations for the domain. Participants recorded their judgment as "yes" or "no".

5.4 Results

A total of 60 Gherkin specifications were generated from food-safety legal provisions – 30 by Claude and 30 by Llama. Specifications were evaluated by 10 participants, each of whom independently assessed 12 specifications (6 per LLM) across five quality criteria: *Relevance*, *Completeness*, *Clarity*, *Singularity*, and *Time Savings* (Table 5.1). The first four of these criteria originate from existing BDD scenario quality frameworks (Section 5.8.2) and are adapted for specification-level evaluation. The last criterion, *Time Savings*, is proposed by us to evaluate participants' overall perception of how useful the generated specifications are. In addition to scale ratings, participants also provided qualitative feedback on the specifications. No participant evaluated outputs from both LLMs for the same legal provision. Each specification was examined by two different participants (forming five distinct participant pairs), ensuring paired evaluations across all 60 specifications and resulting in 120 individual assessments. All participants completed their tasks in full, and no data points were discarded. The following subsections present the evaluation results, beginning with a descriptive analysis and subsequently addressing the research questions.

Table 5.1: Definitions and corresponding scales for the quality criteria evaluated by participants.

Criterion	Definition	Scale	Scale Description
Relevance	Specification matches the intended system behaviour as described in the legal provision while focusing on what is directly pertinent. Only pertinent details for the specified behaviour are presented, i.e., nothing is said that is orthogonal to the behaviour.	**Fully irrelevant**	The specification does not address the requirement at all; it is entirely off-topic.
		Mostly irrelevant	Only a small portion of the specification addresses the requirement, while most content is off-topic.
		Partly relevant	A significant portion of the specification relates to the requirement, but just as much is off-topic.
		Mostly relevant	Most of the specification addresses the requirement, with only a small amount being off-topic.
		Fully relevant	All content in the specification addresses the requirement, with no off-topic information.
Completeness	Specification covers all intended functionality and characteristics, i.e., all functionality and characteristics implied by the legal provision are included without omissions.	**Fully incomplete**	No required information is present; the specification omits all relevant content from the requirement.
		Mostly incomplete	Only a small portion of the required information is present; most relevant content is missing.
		Somewhat incomplete	A significant portion of the required information is present, but an equally significant portion is missing.
		Mostly complete	Most required information is present, with only minor omissions.
		Fully complete	All required information is present; no relevant content is omitted.
Clarity	Specification is written in a clear, precise, and unambiguous manner. Each statement can be uniquely interpreted, avoiding vagueness or misleading language. Technical jargon is used only as necessary.	**Fully unclear**	No part of the specification is understandable; content is highly vague, ambiguous, or misleading.
		Mostly unclear	Only a small portion of the specification is understandable, while the majority is vague, ambiguous, or misleading.
		Partly clear	A significant portion of the specification is understandable, but an equally large portion remains vague, ambiguous, or misleading.
		Mostly clear	Most of the specification is understandable, with only a small portion that is vague, ambiguous, or misleading.
		Fully clear	Every part of the specification is easy to understand, with no vague, ambiguous, or misleading content.
Singularity	Each scenario focuses on a single purpose, i.e., context, event, and outcome work together to describe that purpose. The specification avoids mixing multiple purposes within a single scenario and splits scenarios when distinct functions or events serve different purposes, improving testability and maintainability.	**Fully mixed**	Every scenario combines multiple purposes, making testing and maintenance difficult.
		Mostly mixed	Most scenarios combine multiple purposes, with only occasional separation; testing and maintenance remain difficult.
		Partly singular	A balanced mix: many scenarios focus on a single purpose, but just as many combine multiple purposes.
		Mostly singular	Most scenarios focus on a single purpose, with only a few combining purposes; testing and maintenance are somewhat easier.
		Fully singular	All scenarios focus on a single purpose, making testing and maintenance straightforward.
Time Savings	The extent to which the specification reduces the time a participant would otherwise spend creating the specification from scratch. Reflects how much of the specification can be reused or adapted instead of fully rewritten.	**No time saved**	Does not reduce the workload at all; the specification must be entirely rewritten from scratch.
		Minimal time saved	Only a small fraction of the specification can be reused; most parts must be rewritten from scratch.
		Moderate time saved	A substantial portion of the specification can be reused, but an equally substantial portion still needs to be rewritten from scratch.
		Significant time saved	Most of the specification can be reused; only a small portion must be rewritten from scratch.
		Maximum time saved	Completely eliminates manual work; the specification can be adopted virtually as-is.

5.4.1 Descriptive Analysis of Data

We analyzed the input legal provisions and their corresponding Gherkin specifications by comparing token counts. Figure 5.3 presents a scatter plot of token counts for legal provisions vs. Gherkin specifications generated with both Llama and Claude. The plot also includes the fitted regression lines to highlight the relationship between the measures. Figure 5.4 uses boxplots to compare the distributions of token counts in legal provisions and Gherkin specifications (Llama and Claude combined): legal provisions average ≈ 140.66 tokens (median $= 110$), whereas Gherkin specifications average ≈ 273.01 tokens (median $= 235$). Llama-generated Gherkin specifications (median ≈ 212.5, mean ≈ 265.1 tokens) were shorter than those from Claude (median $= 244$, mean ≈ 285.5 tokens). Nevertheless, the difference was not statistically significant (Wilcoxon rank-sum, $p = 0.18$).

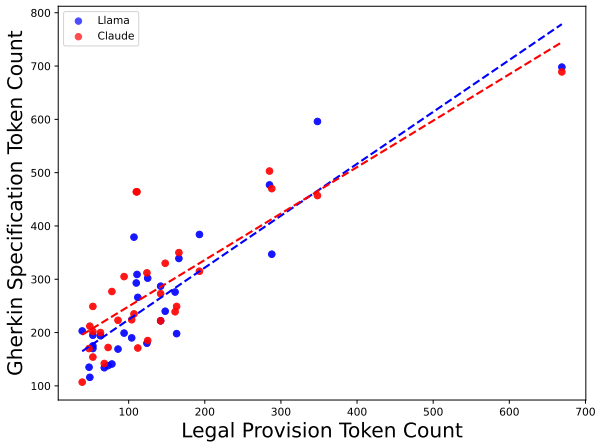


Figure 5.3: Scatter plot of legal-provision token counts versus corresponding Gherkin specification token counts for Llama and Claude, with fitted regression lines.

We further examined the structure of the Gherkin specifications by analyzing the number of steps per scenario. As shown in Fig. 5.5, half of all scenarios contain between three and five steps (median ≈ 4; mean ≈ 3.95), with only a handful of high-end outliers extending to nine or ten steps.

Finally, to understand the granularity of each step, Fig. 5.6 presents a boxplot of step lengths measured in tokens. Typical steps span 9–15 tokens (mean ≈ 12.26, median ≈ 11). Occasional outliers extend beyond these ranges.

5.4.2 Answers to RQs

Participants' ratings are provided on an ordinal scale. We report stacked bar plots for each evaluation criterion and compare participant- and model-level performance using statistical significance tests.

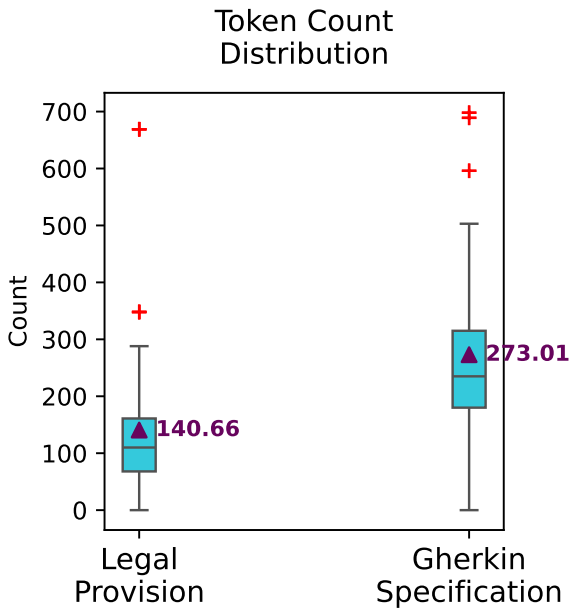


Figure 5.4: Boxplots of the token-count distributions for legal provisions and Gherkin specifications.

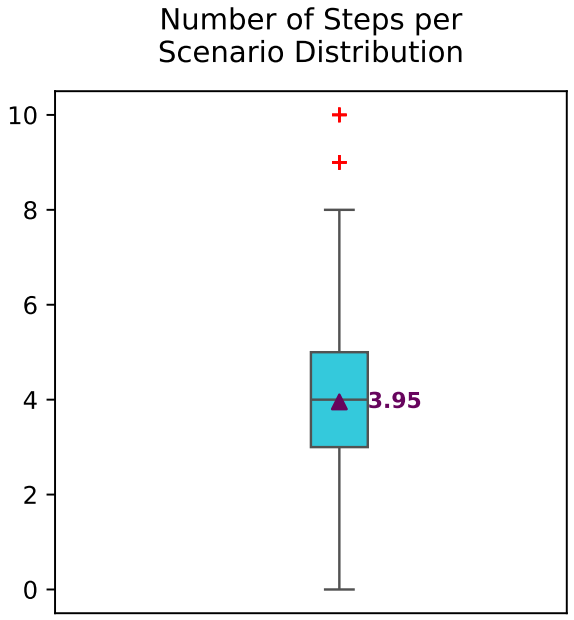


Figure 5.5: Boxplot of the distribution of steps per scenario.

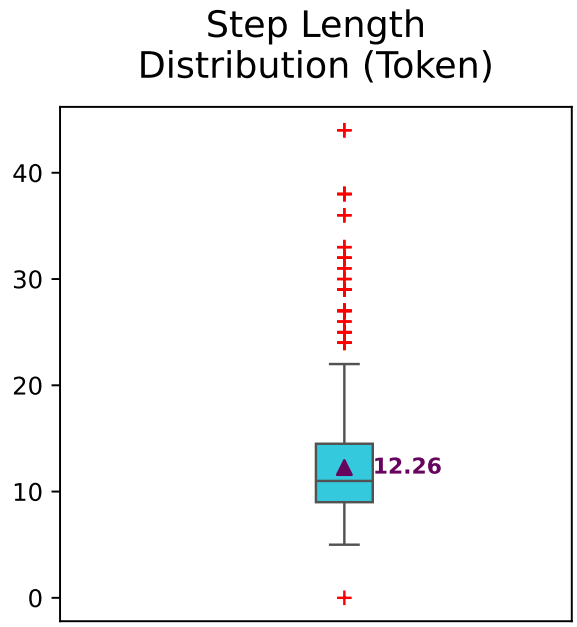


Figure 5.6: Distribution of Gherkin step lengths (in tokens).

RQ1: How do LLMs perform when generating Gherkin specifications from food-safetylegal provisions?

In Fig. 5.7, stacked bar plots display the distribution of participants' ratings across evaluation criteria. The combined results are used to address RQ1. Each bar in the combined results represents 120 ratings for an individual criterion (6 tasks per model $\times$ 2 models $\times$ 10 participants). Overall, the plots indicate a strong consensus among participants' ratings, with all criteria consistently receiving the highest possible median scores ("Fully clear", "Fully singular", "Completely relevant", "Fully singular", and "Maximum time saved"). Several observations emerge from the plots.

1) *Relevance:* The distribution of the ratings for *Relevance* indicates highly relevant Gherkin specifications to legal provisions. Of the 120 ratings, 75% (90) are "Fully relevant" and 20% (24) are "Mostly relevant". Only 2.5% (3) fall into "Partly relevant" and 2.5% (3) into "Mostly irrelevant", indicating a small portion of off-topic information. No responses fall into the "Fully irrelevant" category.

2) *Clarity:* The distribution of the ratings for *Clarity* indicates highly clear Gherkin specifications. Of the 120 ratings, 90% (108) are "Fully clear" and 10% (12) are "Mostly clear", indicating a small fraction of specifications are ambiguous. No responses are "Partly clear", "Mostly unclear", or "Fully unclear", indicating the absence of substantial portions with confusion or difficulty in interpretation.

3) *Singularity:* The distribution of the ratings for *Singularity* indicates single-purpose scenarios in most Gherkin specifications. Of the 120 ratings, 81.7% (98) are "Fully singular", meaning each scenario targets exactly one purpose. 11.7% (14) are "Mostly singular",

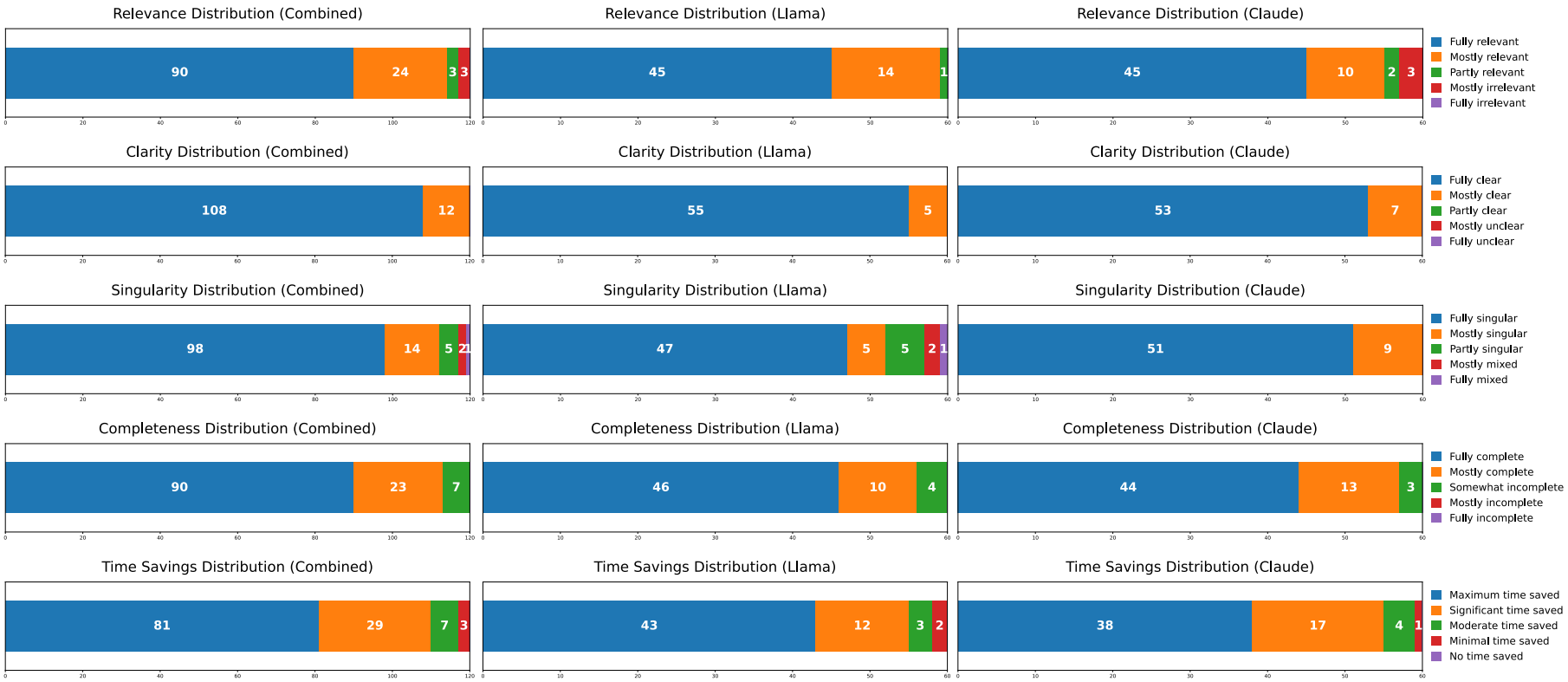


Figure 5.7: Stacked bar plots per criterion for all evaluations across both LLMs combined and stacked bar plots per criterion per LLM.

indicating that while most scenarios maintained clarity of purpose, few scenarios contain multiple purposes. Only 4.2% (5) are "Partly singular", 1.7% (2) are "Mostly mixed", and 0.8% (1) "Fully mixed", indicating very few scenarios contain overlapping purposes.

4) *Completeness:* The distribution of the ratings for *Completeness* indicates that most Gherkin specifications capture the intended functions and characteristics implied by the legal provisions without omissions. Of the total 120 ratings, 75% (90) are "Fully complete", 19.2% (23) as "Mostly Complete", reflecting specifications with a small portion of relevant information being omitted. Only 5.8% (7) are "Somewhat incomplete", and none are "Mostly incomplete" or "Fully incomplete", indicating most specifications include at least a substantial portion of the necessary information.

5) *Time Savings:* The distribution of the ratings for *Time Savings* indicates substantial efficiency gains provided by Gherkin specifications. Of the 120 ratings, 67.5% (81) are "Maximum time saved", reducing manual effort by eliminating extensive rewriting. 24.2% (29) are "Significant time saved", showing small portions require rewriting. 5.8% (7) are "Moderate time saved", showing substantial portions need editing. Only 2.5% (3) are "Minimal time saved", indicating only small portions are reusable. No responses are "No time saved", thus there is some efficiency gain in all cases.

We recall from our study design (Section 5.3.3) that each participant pair correspond to two participants who evaluate the same set of 12 Gherkin specifications. With ten participants, this yields five independent pairs. For each criterion, the two participants in a pair provide separate ratings, producing two sets of 12 ratings per pair for comparison. We use the Wilcoxon signed-rank test [41] to compare paired ratings within participant pairs (matched by specification); this test is a suitable choice for paired ordinal data without assuming a normal distribution [50, 129]. To control the false discovery rate (FDR) when performing multiple comparisons, we apply the Benjamini–Hochberg (BH) correction [27].

The p-values (and BH-adjusted p-values) in Table 5.2 test the null hypothesis (H_0) that there is no systematic within-specification shift between the paired ratings (i.e., the paired changes are centred at zero). With two exceptions, none of the comparisons across the five criteria – *Relevance, Completeness, Clarity, Singularity,* and *Time Savings* – shows a statistically significant difference. Specifically, participant pairs 2 and 3 yield nominally significant unadjusted p-values for *Time Savings*; however, these effects do not survive BH correction. Thus, after BH adjustment, we find no statistically reliable differences for any quality criterion.

Across all five quality criteria – *Relevance, Completeness, Clarity, Singularity,* and *Time Savings* – none of the comparisons shows a statistically significant difference.

Regarding the plausibility of the specifications, we recorded four instances in which participants indicated that a generated specification was "not plausible," with one case shared between two participants in a pair; we later present this case as Example 1 in Section 5.5. Therefore, there were only three unique specifications deemed implausible by participants. The assessment of plausibility was based on common sense and basic physical reasoning.

Table 5.2: Wilcoxon signed-rank tests for each participant pair (two raters who scored the same 12 specifications) across the five quality criteria. Cells report p-value, and Benjamini-Hochberg-adjusted p-values (p_{BH}).

	Relevance		Completeness		Clarity		Singularity		Time Savings	
	p-value	p_{BH}-value	p-value	p_{BH}-value	p-value	p_{BH}-value	p-value	p_{BH}-value	p-value	p_{BH}-value
user pair 1	0.125	0.446	0.125	0.446	1	1	1	1	0.125	0.446
user pair 2	1	1	1	1	1	1	0.125	0.446	0.03	0.375
user pair 3	0.06	0.446	0.45	0.962	1	1	0.5	0.962	0.03	0.375
user pair 4	1	1	0.37	0.925	0.5	0.962	0.25	0.694	1	1
user pair 5	1	1	1	1	1	1	1	1	0.22	0.688

The answer to **RQ1** *is: LLM-generated Gherkin specifications received predominantly high ratings from participants across all evaluation criteria. Specifically, for* Relevance*, 95% of ratings are either "Fully relevant" (75%) or "Mostly relevant" (20%), with no "Fully irrelevant" ratings. Regarding* Clarity*, participants rated the specifications highly, with 90% as "Fully clear" and 10% as "Mostly clear", with no substantial ambiguity reported. In terms of* Singularity*, 93.4% of ratings are positive ("Fully singular", 81.7%; "Mostly singular", 11.7%), while 6.7% of specifications contain a substantial number of multiple-purpose scenarios. For* Completeness*, 94.2% of ratings are either "Fully complete" (75%) or "Mostly complete" (19.2%), signifying comprehensive coverage of legal requirements, with no ratings falling into the bottom two rating categories. Lastly,* Time Savings *received 67.5% "Maximum time saved" and 24.2% "Significant time saved" ratings, highlighting strong perceived time-efficiency, with no "No time saved" ratings. Further, no statistically reliable differences were observed across participants. Only three cases of non-plausibility were identified, suggesting that almost all specifications are considered to be realistic.*

RQ2: How do Llama and Claude compare when generating Gherkin specifications from food-safety legal provisions?

We use the per-LLM results in Fig. 5.7 to answer RQ2, presenting a direct model-to-model comparison across the evaluation criteria. Each bar in the per-LLM results reflects 60 ratings per model per criterion (6 tasks $\times$ 10 participants), highlighting differences in performance between Llama and Claude. Several observations emerge from the plots.

1) *Relevance:* Both Llama and Claude perform well in generating relevant Gherkin specifications, with similar distributions of participant ratings. For Llama, 45 responses (75%) are "Fully relevant", followed by 14 responses (23.3%) as "Mostly relevant", and 1 response (1.7%) as "Partly relevant". There are no ratings in the lowest two categories. Claude also achieves 45 "Fully relevant" ratings (75%), with 10 (16.7%) as "Mostly relevant", 2 (3.3%) as "Partly relevant", and 3 (5%) as "Mostly irrelevant". As with Llama, no responses are rated as "Fully irrelevant". While both models' outputs show high alignment with the legal provisions, Llama's output distribution is slightly more concentrated at the

top, with fewer mid- or low-range responses than Claude. These results suggest that both LLMs are highly capable of producing relevant content to the legal provision, though Llama appears to have slightly better consistency in maintaining top-tier relevance.

2) *Clarity:* Both Llama and Claude received high ratings for clarity, with no responses falling into the bottom three categories. For Llama, 55 out of 60 ratings (91.7%) are "Fully clear", indicating that nearly all specifications are perceived as unambiguous. The remaining 5 responses (8.3%) are "Mostly clear", suggesting ambiguity in a few cases. Claude's ratings are similarly strong, with 53 responses (88.3%) as "Fully clear" and 7 (11.7%) as "Mostly clear". While both models are able to generate consistently clear output, Llama shows a slight advantage, with a greater proportion of top-tier ratings.

3) *Completeness:* Both Llama and Claude received strong ratings for completeness. For Llama, 46 responses (76.7%) are "Fully complete", with an additional 10 (16.7%) as "Mostly complete". Only four responses (6.7%) are "Somewhat incomplete". Claude shows comparable performance with 44 responses (73.3%) as "Fully complete", 13 (21.7%) as "Mostly complete", and 3 (5%) as "Somewhat incomplete". Neither model received ratings in the two bottom categories. Llama offers marginally more coverage of legal content as presented by "Fully complete" category (76.7% vs. 73.3%), although the distributions are close.

4) *Singularity:* Claude and Llama both perform well in maintaining scenario singularity, though their rating distributions show differences. Claude received 51 ratings (85%) as "Fully singular" and 9 (15%) as "Mostly singular". No responses are rated as "Partly singular", "Mostly mixed", or "Fully mixed". Llama received 47 "Fully singular" ratings (78.3%) and 5 "Mostly singular" ratings (8.3%). However, it also received more dispersed ratings with 5 (8.3%) as "Partly singular", 2 (3.3%) as "Mostly mixed", and 1 (1.7%) as "Fully mixed". Overall, Claude shows more consistent adherence to the principle of scenario singularity.

5) *Time Savings:* Claude and Llama are effective in reducing the effort required to produce specifications from scratch, though their distribution differs slightly. For Llama, 43 responses (71.7%) are "Maximum time saved", suggesting that in most cases, the specifications could be reused with minimal editing. An additional 12 responses (20%) are "Significant time saved", reflecting significant but not full reuse. Only 3 responses (5%) are "Moderate time saved" and 2 (3.3%) are "Minimal time saved". Claude showed slightly lower top-tier scores with 38 responses (63.3%) as "Maximum time saved" and 17 (28.3%) as "Significant time saved". Only 4 responses (6.7%) are "Moderate time saved", and 1 response (1.7%) as "Minimal time saved". Overall, both models offer considerable time savings, with Llama slightly ahead in producing outputs that participants could adopt almost as-is.

We apply the Mann-Whitney U Test [41] to compare the outputs of Llama and Claude across our five quality criteria. The null hypothesis (H_0) for each comparison states that there is no difference in the distribution of participant ratings between the two models for a given quality criterion. The results in Table 5.3 show that no statistically significant differences are observed between Llama and Claude on any quality criterion, under either the unadjusted or BH-adjusted p-values. These findings indicate that Llama and Claude

Table 5.3: Mann-Whitney U tests comparing Llama and Claude across the five quality criteria. Cells report p-value, and Benjamini-Hochberg-adjusted p-values (p_{BH}).

	Relevance		Completeness		Clarity		Singularity		Time Savings	
	p-value	p_{BH}-value	p-value	p_{BH}-value	p-value	p_{BH}-value	p-value	p_{BH}-value	p-value	p_{BH}-value
Llama vs. Claude	0.87	0.87	0.8	0.87	0.75	0.87	0.41	0.87	0.48	0.87

perform comparably across all five evaluated quality criteria, with no evidence of either model outperforming the other in our study context.

The answer to **RQ2** *is: Claude and Llama perform comparably well across all evaluation criteria when generating Gherkin specifications from legal provisions. Specifically, for* Relevance*, both models received identical proportions of top-tier ratings ("Fully relevant") at 75%, although Claude has slightly more lower-end responses. In terms of* Clarity*, both models excel, with Llama slightly outperforming Claude, achieving 91.7% "Fully clear" ratings compared to Claude's 88.3%. For* Completeness*, Llama achieved a marginally higher proportion of "Fully complete" responses (76.7%) versus Claude (73.3%), with both models consistently covering functionalities and characteristics of legal provision. Regarding* Singularity*, Claude is more consistent with 85% of responses as "Fully singular" compared to Llama's 78.3%, and Llama shows slightly more variability with mixed-purpose scenarios. Lastly, for* Time Savings*, Llama produced marginally higher top ratings ("Maximum time saved", 71.7%) compared to Claude (63.3%), though both still provided substantial efficiency gains. Statistical analysis (Mann-Whitney U Test) found no significant differences between Llama and Claude for any criterion.*

5.5 Analysis of Qualitative Feedback

To complement the quantitative results presented in Section 5.4, we now turn to a *qualitative* analysis of the textual feedback provided by participants in order to gain insights into the types of issues that arise and the reasoning behind their judgments. In this section, we first summarize descriptive statistics on the comments received, then present the key themes that emerged from the analysis, and finally illustrate the most salient themes with representative examples.

5.5.1 Summary Statistics

Across 120 assessments, participants left a total of 105 comments. Of these, 64 (61%) pointed out potential issues in the generated specifications, while 41 (39%) were approvals or alternative suggestions, even though all criteria had been assigned the highest score. Thus, nearly two thirds of the feedback identified specification issues.

Breaking statistics down per participant, each participant contributed between 5 and 12 comments each (mean = 10.5, SD = 2.6), with seven participants providing the maximum

of 12 comments. When filtering out non-issue comments (approvals or suggestions), the number of issue-specific comments per participant ranged from 3 to 11 (mean = 6.4, SD = 2.6), with one participant providing the maximum of 11 unique issue-specific comments. The number of approval comments ranged from 1 to 8 per participant (mean = 4.1, SD = 2.7), with two participants providing the maximum of 8 approval comments. Every one of the 10 participants provided both issue-specific comments and approval comments.

Aggregated by quality criterion, we obtained 39 comments related to *Time Savings*, 30 to *Relevance*, 30 to *Completeness*, 22 to *Singularity*, and 12 to *Clarity*. *Clarity* received input from 7 unique participants, while the other criteria received feedback from 9 unique participants each.

5.5.2 Recurring Themes in Participant Comments

We examined the textual feedback from participants through a close reading of all comments and the identification of recurring themes, which revealed common sources of concern such as irrelevant details, omissions, and ambiguities. The main themes that emerged pertained to three quality criteria: *Relevance*, *Completeness*, and *Singularity*.

For *Relevance*, participants noted issues such as off-topic or hallucinated content not grounded in the legal text, unnecessary repetition of information, and misinterpretations where the model generated inaccurate statements. Overall, our analysis shows that generated specifications sometimes contained unnecessary assumptions about system behaviours. Observed patterns further indicated that scenario creation was at times influenced by constraints and practices not present in the source, which affected the perceived *Relevance* of the specifications.

For *Completeness*, the most common concerns were omissions of entire requirements, omissions of specific clauses, or omissions of details. Overall, our analysis shows that generated specifications on occasion omitted necessary details and phrases. In some cases, functions and characteristics supported by the legal provision were not clearly reflected in the specification. These omissions contributed to misinterpretations and increased participants' effort in rewriting, which negatively affected the *Completeness*.

Finally, for *Singularity*, when legal provisions were lengthy or covered multiple obligations (e.g., size, mass, or temperature measurements), participants identified situations where scenarios were not split, making it harder to preserve a single purpose. Conversely, some comments pointed to unnecessary scenario splitting, where combined scenarios would have been better. Overall, our analysis shows that when legal provisions were long and complex, generated specifications sometimes contained multiple purposes within the same scenario, which reduced *Singularity* and also complicated *Clarity*. In other cases, excessive splitting created scenarios that were fragmented and less comprehensible. Both tendencies introduced ambiguity about how the specification related to the legal text, increasing the risk of misinterpretation.

Figure 5.8 presents a breakdown of participant feedback on *Relevance*, *Completeness*, and *Singularity*, along with the frequency of each identified theme. For each quality criterion, the figure also includes a "general feedback" category for comments that did not

address a specific deficiency. In the remainder of this section, we illustrate some of the themes from Fig. 5.8 using three legal provisions, their corresponding Gherkin specifications, and selected participant feedback.

Relevance (30 comments)

Off-topic or hallucinated content (16/30): Gherkin specification describes assumptions, functions, or characteristics not expressly stated or implied by the legal provision, and therefore lacking explicit or implicit relevance to it.

Redundancy (6/30): Gherkin specification repeats information unnecessarily, reducing overall relevance.

Misinterpretation (5/30): Gherkin specification contains mistakes resulting from the LLM's misinterpretation of a legal provision.

General feedback (3/30): General suggestions for improvement that do not fit into the above three specific issue categories.

Completeness (30 comments)

Omission of requirements (8/30): Some legal requirements have been omitted in their entirety from the Gherkin specification.

Omission of clauses (13/30): Some steps entailed by the legal provision are absent from the Gherkin specification.

Omission of variants (4/30): Gherkin specification fails to cover all product variants (e.g., different types of meat) or action variants (e.g., import, export, trade).

General feedback (5/30): General suggestions for improvement that do not fit into the above three specific issue categories.

Singularity (22 comments)

Mixed purposes in a single scenario (14/22): Some scenarios combine multiple purposes, making testing and maintenance difficult.

Unnecessary scenario splitting (6/22): Scenarios have been unnecessarily split when they could have been combined.

General feedback (2/22): General suggestions for improvement that do not fit into the two specific categories above.

Figure 5.8: Issue themes observed in LLM-generated Gherkin specifications, grouped by quality criteria. Numbers in parentheses indicate the counts of participant comments. An individual comment could relate to multiple quality criteria if it addressed more than one type of issue simultaneously.

```
Feature: Liquid Food Product Inspection

Scenario: Measuring net quantity of prepackaged liquid food
Given a consumer prepackaged food product containing liquid
And the liquid temperature is stabilized at 20°C
When the inspector measures the net quantity
Then the system should record the measurement value
And store the measurement with the temperature condition

Scenario: Attempting measurement at incorrect temperature
Given a consumer prepackaged food product containing liquid
And the liquid temperature is not at 20°C
When the inspector attempts to measure the net quantity
Then the system should display a warning message
And prevent the measurement from being recorded
And prompt for temperature adjustment to 20°C
```

Figure 5.9: LLM-derived Gherkin specification for Example 1.

The first example, Example 1, focuses on the issues observed in relation to *Relevance*:

Example 1 – Legal Provision

(5) In the case of an inspection of a consumer prepackaged food that consists of a liquid, the net quantity of the food must be determined on the basis of the assumption that the liquid is at a temperature of 20 °C.

Figure 5.9 shows an LLM-derived Gherkin specification for Example 1. The *Relevance*-related feedback from the two participants who examined this specification is as follows.

Feedback (Example 1) – Participant 1

I think that the second scenario is irrelevant to the requirement content as it relies on assumptions about the system (such as assuming it will display a warning). However, it's unclear whether the system even includes a visual i/o component.

Feedback (Example 1) – Participant 2

I find the content of the second scenario to be irrelevant and include hallucinations as the legal requirements does not talk about displaying a warning. Also, the assumption that all liquids with prepackaged food can be held at exactly 20 degrees is far-fetched. The concept of matching the volume in current temperature to what it should be at 20 degrees is much more feasible.

Here, both participants critique the second scenario. One argues that requiring the system to display a warning message and prevent measurement goes beyond what the provision could reasonably entail, and should therefore be discarded to improve *Relevance*.

The second participant observes that the specification should reflect what is plausible in real-world practice, noting that physically bringing a liquid to a temperature of 20 °C merely for the purpose of measuring its volume is impractical. Instead, they conclude that the most reasonable interpretation is that the volume should be (mathematically) adjusted as if the liquid were at 20 °C. Based on the feedback received, a revised Gherkin specification, which more accurately reflects realistic measurement practices, is presented in Fig. 5.10.

```
Feature: Liquid Food Product Inspection

As a food safety inspector,
I need to measure the net quantity of prepackaged liquid food
adjusted to 20°C
So that I can ensure accurate quantity measurements for consumer
protection

Background: Given the inspection system is calibrated for
temperature-adjusted liquid measurements
And the system has temperature monitoring and adjustment
capabilities

Scenario: Measuring net quantity of prepackaged liquid food
Given a consumer prepackaged food product containing liquid
And the liquid temperature is measured at its current state
When the inspector measures the net quantity
Then the system should calculate the adjusted volume as if the
liquid is at 20°C
And record the adjusted measurement value
And record the temperature used for adjustment
```

Figure 5.10: Revised Gherkin specification for Example 1.

Our second example, Example 2, concerns issues observed in relation to *Completeness*. The feedback from participants on the LLM-derived Gherkin specification for Example 2, shown in Fig. 5.11, indicates that several omissions are present.

Example 2 – Legal Provision

223 (1) If a consumer prepackaged food was wholly manufactured, processed or produced in a foreign state and the name and principal place of business of the person in Canada for whom it was manufactured, processed or produced or the person by whom it was stored, packaged or labelled in Canada is shown on its label, that information must be preceded by the expressions “Imported by” and “importé par” or “Imported for” and “importé pour”, as the case may be, unless the geographic origin of the consumer prepackaged food is shown on the label in accordance with subsection (3).

```
Feature: Foreign Manufactured Prepackaged Food Labeling
Rule: Foreign manufactured food must display either Canadian
handler information or geographic origin

Background: Given a consumer prepackaged food product exists
And it was wholly manufactured in a foreign state

Scenario: Displaying Canadian Handler Information
Given a foreign manufactured prepackaged food product
When geographic origin is not shown on the label per subsection
(3)
Then the label must display the following Canadian handler
information:

| Information Type  | Requirement  | Name   | Person responsible
in Canada |  Principal place of business  | Canadian business
location    |

Scenario: Geographic Origin Display Exception
Given a foreign manufactured prepackaged food product
When the geographic origin is shown on the label
And the origin information complies with subsection (3)
Then Canadian handler information is not required on the label
```

Figure 5.11: LLM-derived Gherkin specification for Example 2.

Feedback (Example 2) – Participant 1

The "import" expression listed in the requirement did not appear to have a scenario associated to it. I've provided my version of this new scenario.

Feedback (Example 2) – Participant 2

I find that the generated specification misses what the requirement was discussing and interprets it incorrectly. Based on my understanding the requirement is discussing when "imported for" or "imported by" should be a "label data". But the generated specification has a scenario that does not mention this "label data" and focuses on if the name of the person should be a "label data". The specification would be more complete if it contained all terms "manufactured", "processed" or "produced". I didn't find ambiguity other than what was mentioned.

Taken together, the feedback suggests that the LLM-derived specification is missing some important legal terms and fails to capture the intended labelling requirements. In particular, the specification does not adequately reflect the mandatory bilingual expressions or the conditions under which they apply. To address the incompleteness, a revised version that more faithfully represents the provision is presented in Fig. 5.12.

Our third and final example illustrates issues related to *Singularity*, particularly the tendency of LLMs to generate multi-purpose scenarios. Due to their length, the underlying legal provision, the LLM-derived Gherkin specification, and the revised Gherkin specification are provided in Appendix C. The feedback on *Singularity* from the two participants who reviewed the LLM-derived Gherkin specification (Fig. C.1) is as follows:

Feedback (Example 3) – Participant 1

Response has stacked all of the sub-requirements into two specifications, per equine and per bird. Many details have been omitted by the Gherkin for the table. The answer is totally wrong. A Detailed version is provided.

Feedback (Example 3) – Participant 2

I can add similar elements as the background and write different parts in each scenario. The main problem I had with this one is that (h) is only talking about equine (not bird). But the second scenario has included subitems of (h) for a bird, which is incorrect.

The main problem with this example is that the LLM-derived Gherkin specification aggregates all sub-requirements into two scenarios, each with a large "Then" step including an information table, rather than producing one scenario per individual obligation. For example, sub-clauses (e)(i), (e)(ii), and (e)(iii) concerning birds should each appear as separate scenario when they represent distinct checks.

In addition to this primary problem with *Singularity*, the LLM-derived specification also

```
Feature: Labelling of foreign-manufactured consumer pre-packaged
food

Background: Given a consumer pre-packaged food product was
wholly manufactured, processed, or produced in a foreign state

# 1. Mandatory bilingual ``Imported by / Imported for'' prefix

Scenario: Canadian responsible-party information is shown and
geographic origin is NOT shown
Given the label shows the name and principal place of business
of a person in Canada who is responsible for the product
And the geographic origin of the food is not shown on the label
in accordance with subsection(3)
Then that Canadian responsible-party information must be
preceded by one of the following bilingual phrases:

| English    | French        |
| Imported by  | importé par    |
| Imported for | importé pour   |

# 2. Exception when geographic origin is shown

Scenario: Canadian responsible-party information is shown and
geographic origin IS shown
Given the label shows the name and principal place of business
of a person in Canada who is responsible for the product
And the geographic origin of the food is shown on the label in
accordance with subsection(3)
Then the phrases ``Imported by/importé par'' or ``Imported
for/importé pour'' are not required
```

Figure 5.12: Revised Gherkin specification for Example 2.

entirely omits several nested items. For instance, with respect to birds, it fails to include recovery dates for diseases, extra-label drug prescription attestations, and other details specified under clause (g) sub-items. These missing elements fall under the *Completeness* criterion. Since similar omissions were already illustrated in Example 2, we do not elaborate further here.

As one possible revision based on the participants' feedback, Figs. C.2 and C.3 in Appendix C present a case where the requirements for equines and birds are separated into two distinct feature files, each containing single-purpose scenarios.

5.6 Discussion

In this section, we discuss the criticality of LLM generation errors for the task investigated in this article as well as the broader implications of our findings.

5.6.1 LLM Failure Risks

The issues observed in our qualitative analysis of LLM-generated Gherkin specifications, namely missing clauses, hallucinated content, and conflated intents, carry different levels of risk. In particular, in safety-critical contexts such as food handling, omissions appear to be the most consequential risk: unless caught by humans, the LLM overlooking a regulatory constraint can lead to potentially unsafe outcomes that nonetheless pass testing, only to fail under real-world regulatory scrutiny or operational conditions. In contrast, hallucinated content, although still serious, is easier for human reviewers to identify, given the expectation that LLMs are prone to generating spurious content. Multi-purpose scenarios, while undesirable, are secondary to omissions and hallucinations, as they are likely to be naturally identified and resolved during downstream refinement activities. The *main takeaway* here is that automated derivation of Gherkin specifications from laws and regulations – like most other LLM-based automation – cannot be assumed to be fully accurate, with omissions and hallucinations remaining notable risks. Educating users about these risks is a key mitigation measure; for the same reason, human-in-the-loop verification is especially important in safety-critical domains where non-compliance could lead to harm to individuals, the environment, or society at large.

5.6.2 Implications for Research

Our study provides empirical evidence that existing LLMs can support the transition from laws and regulations to behavioural specifications in a developer-centric format. These results broaden the feasible scope of automation for legal requirements and suggest that legal texts can be used not only for verbatim extraction or classification, but also as direct sources for structured software engineering artifacts. Methodologically, our study provides a reusable blueprint for evaluating LLM-generated artifacts using human judgment. At the same time, our qualitative findings highlight important limitations that should be

addressed in future research. The issues discussed in Section 5.6.1 point to the need for techniques that (i) ensure faithfulness to the source provisions, (ii) support systematic decomposition of long or cross-referenced clauses into reviewable units, and (iii) make the transformation process more explainable. This motivates research on prompt and glossary design as well as human-AI review protocols, including critique-refine loops, tailored to compliance-driven requirements analysis.

5.6.3 Implications for Practice

Overall, and while strictly scoped to our study domain (as is typical for case-study research), our results suggest that LLMs at their current level of maturity can serve as effective co-authors of behavioural specifications in regulated domains, especially as a *first-draft* mechanism. High ratings across the quality criteria, together with strong perceived time savings, indicate that teams should be able to start from LLM-generated specifications rather than writing scenarios from scratch. In practice, this shifts effort towards review and traceability management, rather than initial authoring. However, practitioners should assume that LLM outputs are *draft* artifacts rather than authoritative statements of obligations. Several participant comments flagged issues (Section 5.5). Consequently, effective adoption should be framed as a human-in-the-loop workflow: (i) scope and, where necessary, decompose provisions into smaller fragments; (ii) build a domain glossary to establish terminology; (iii) generate candidate specifications using controlled prompts; and (iv) review outputs against a checklist reflecting relevance, completeness, clarity, and single-purpose scenario structure, complemented by a plausibility check for unrealistic behaviours. Teams should keep legal inputs, prompts, and generated outputs under configuration management to support auditability and reproducibility, and apply extra scrutiny to lengthy or cross-referenced provisions, where the risk of omissions is higher. When outputs contain implausible behaviour or hallucinations, teams should treat this as a signal to rescope the input or revise the prompting/glossary strategy, not merely as a one-off defect to patch.

5.7 Validity Considerations

Below, we discuss validity considerations for our study, focusing on internal, external, construct, and statistical conclusion validity, and describing the measures we took to mitigate threats.

Internal validity. An important threat to internal validity is evaluator bias. Although participants did not know which LLM had produced the outputs they reviewed, they were aware they were evaluating LLM-generated text and could adjust their ratings – consciously or unconsciously – accordingly. We mitigated this risk through standardized training, detailed scoring rubrics, and illustrative examples to calibrate judgments before any evaluations began.

A second threat is that participants may not have fully considered downstream consequences when judging the specifications. Participants understood that food handling is a

safety-critical domain, but we did not overstress potential real-world harms from errors, since doing so could have shifted participants away from their natural decision-making. This creates a trade-off: ratings may reflect how well the specifications match the legal provisions and the predefined quality criteria more than how they would be scrutinized in a consequence-focused compliance setting. To keep evaluations grounded in reality, we included a plausibility check that asked whether the specifications could reasonably materialize in a real-world implementation. We also required brief qualitative feedback when participants identified problems.

A third threat is order-related carryover effects. While each participant rated any given legal provision only once (never comparing outputs from both models for the same provision), early tasks could still anchor their internal scale for later, unrelated provisions. To address this, we randomized task order for each participant and balanced model assignments across their sequences so that exposure to Claude and Llama was evenly distributed rather than clustered. Assigning two independent raters per specification further reduced this risk.

A fourth threat is participant expertise. We deliberately recruited participants with a software engineering background rather than legal or food-safety expertise, because evaluating Gherkin specifications requires software engineering competence (e.g., familiarity with BDD conventions, testability, and scenario structure) that domain experts do not necessarily have. Nonetheless, residual risks remain: although participants were not asked to adjudicate legal intent or enforcement, assessing whether a scenario is faithful to what is stated in a legal provision still requires interpretation, thus creating a risk of misinterpretation. We mitigated this risk by having the author team (who have worked on food safety and the legal framework in scope for over four years and have interacted with food-safety experts) available during the study to provide interpretive assistance with the provisions. Yet, legal or food-safety professionals may identify nuances (e.g., scope, implicit exceptions, term definitions) that software engineers could miss; outcomes may therefore differ with mixed teams, making replication with mixed participants a natural next step.

External validity. Our findings are bounded by our study context: food-safety regulations from North American jurisdictions, 10 student participants from a single university, and two specific LLM versions used with a fixed prompt template. To avoid overgeneralization, we limit our claims to this domain and configuration, and further ensured balanced exposure to both models for all participants. Although all participants came from the same institution, they ranged from senior undergraduates to final-year PhD candidates, providing a diversity of experience levels and reducing the likelihood that our results reflect a single skill level.

In our prompt design, we assigned a consistent "senior requirements engineering expert" persona in the system message across all generations to reduce prompt-induced variability and to reflect a plausible software engineering stakeholder who would consume and refine behavioural specifications. Nevertheless, persona framing can steer an LLM's emphasis and style (e.g., prioritizing testability, completeness, or compliance interpretation), which may in turn influence both the structure and the content of the resulting specifications. Alternative roles such as a quality engineer, compliance officer, legal counsel, or business

analyst could lead to different trade-offs and potentially different outcomes [140, 235].

The provisions in our study are primarily made up of operational requirements (e.g., measurable constraints and concrete obligations). Laws and regulations stated at higher levels of abstraction – for example, those articulating broad principles, rights, or discretionary standards – may require substantial contextual interpretation before they can be operationalized. Such provisions can have intricate cross-references, exceptions, and open terms that do not map as naturally to a `Given-When-Then` pattern.

A further limitation concerns the replicability of our results. Replications with LLMs can be challenging because outputs may vary across runs and over time due to factors such as stochastic generation, provider-side updates, and model-version changes, even when prompts are held constant. Recent work has begun to propose reporting and packaging guidelines to mitigate some of these issues [25]. While such guidelines are still evolving and not yet widely established, we documented our experimental configuration as precisely as possible and provide a replication package to support re-analysis and partial regeneration. *Construct validity.* We evaluated five quality criteria on ordinal scales, plus a binary plausibility check. "Time Savings" is necessarily subjective (self-assessed rather than time-tracked), and the plausibility check is coarse (Yes/No). We constrained drift by supplying concise definitions, clear scale descriptions, and illustrative examples during training, so raters applied shared interpretations when scoring each criterion. Using established BDD-inspired criteria (*Relevance*, *Completeness*, *Clarity*, *Singularity*) further grounded the constructs in prior literature, and we applied the same criteria uniformly across all tasks.

Also, We did not formally validate the quality criteria themselves, or establish that different evaluators would necessarily converge on identical interpretations of each criterion. In particular, criteria such as *Relevance* involve judgment about how a legal provision should be operationalized in software behaviour, which may reasonably vary across evaluators. However, we mitigated subjectivity through explicit definitions, ordinal scales, training examples, and paired evaluations.

Statistical conclusion validity. The sample size (10 participants; 60 specifications; 120 ratings) limits power and introduces some clustering (multiple ratings per participant and provision pairings). To maintain appropriate inference under these constraints, we used non-parametric tests suited to ordinal data, reported p-values, and applied the same analysis workflow across criteria and comparisons.

5.8 Related Work

We review related work in two areas: (1) the use of LLMs in requirements engineering (RE) and specification generation, and (2) evaluating and improving BDD quality.

5.8.1 LLMs in RE and Specification Generation

LLMs are being increasingly used for requirements automation. We categorize prior work into three themes: (i) *analysis and assurance*, (ii) *generation and transformation*, and (iii) *compliance-oriented NLP*.

Analysis and Assurance

LLMs have been used to automate common requirements analysis tasks, including ambiguity detection [22], requirements classification [43, 51, 112, 158], extraction of contractual obligations [128, 191], smell and dependency detection [92, 229], and incompleteness checking [153]. More recently, researchers have applied in-context reasoning to support quality assurance [64] and to perform satisfiability checks on requirements [195].

Generation and Transformation

Research in this area focuses on generating or restructuring RE artifacts. Rahman et al. [186] convert requirement documents into user-story templates, while Ronanki et al. [190] and Görner et al. [79] generate requirements elicitation interview scripts. Nayak et al. [167], Spoletini and Ferrari [206], and Ferrari and Spoletini [65] map natural-language requirements into formal or domain-specific languages (DSLs) suitable for downstream verification. Wei et al. [231] generate code stubs and test scaffolds from high-level requirements descriptions.

Lutze et al. [154] enrich product-concept catalogues and generate tailored specifications for Ambient Assisted Living devices, highlighting prompt-engineering challenges in resource-constrained settings. Xie et al. [237] find that few-shot learning can significantly enhance specification fidelity when LLMs process complex or poorly framed prompts. Ronanki et al. [190] report that requirements generated by LLMs show high atomicity and, in some cases, surpass those produced by human elicitors across several quality criteria. Ferreira et al. [66] generate Gherkin scenarios from user stories, produce executable test cases, and evaluate them through user feedback on scenario usefulness, script usability, and correction effort.

Wang et al. [227] generate acceptance criteria by incorporating multi-modal requirements data, including textual descriptions and visual UI artifacts, and evaluate them along the dimensions of relevance, completeness, understandability, and coverage. Shi et al. [199] refine Gherkin specifications using stakeholder feedback from vision videos. Their results show that LLMs can generate detailed Gherkin specifications, but risk formulation errors that require human validation. Finally, Fernandes de Sousa [205] rewrite informal test cases into Gherkin scenarios. This work reinforces industry's growing interest in AI-assisted BDD, and aligns with our goals of reducing manual specification writing effort.

The above studies focus on transforming user stories, formal DSLs, and artifacts such as code as well as non-textual media like videos. None, however, derive Gherkin specifications from legal provisions.

Beyond LLM-based generation, empirical requirements engineering has a long history of human-centred studies on requirements quality. Mund et al. [166] combine surveys and controlled experiments to examine when requirements quality affects downstream development tasks. Winter et al. [233] use a web-based experiment to study how quantifier formulations influence requirements readability and error rates. More recently, Frattini et al. [69] report a controlled experiment quantifying the effects of passive voice and ambiguous pronouns on domain-model derivation. Methodologically, our quasi-experiment follows this line of work through human-based analysis of requirements-related artifacts, but differs in two key respects: first, we examine LLM-generated Gherkin specifications derived from legal provisions, thus introducing a legal-compliance perspective; second, we focus on BDD-inspired quality criteria rather than modelling accuracy or reading performance, thus more specifically catering to agile and BDD-oriented development practices.

Legal Reasoning and Compliance

Dahl et al. [48] assess LLMs on specific factual questions about federal case law, reporting hallucination rates between 58% and 88%. Similarly, Magesh et al. [155] evaluate commercial retrieval-augmented legal research tools, finding that 17% to 33% of responses contain incorrect legal statements or unsupported citations, risks that have led to real-world sanctions for practitioners [212]. Our work differs fundamentally by focusing on a structurally distinct, syntax-based translation, mapping legal provisions to Gherkin specifications, rather than the open-world knowledge retrieval and legal reasoning that often leads to these errors.

Singhal et al. [200] take a code-centric approach by prompting LLMs to generate Python code that instantiates a domain-specific metamodel of legal constructs (Sections, Rules, Conditions, References). The Python code is executable and derived from statutory text via textual entailment and in-context learning. In contrast, our work is concerned with generating and evaluating the quality of Gherkin specifications from legal texts.

Kesari et al. [138] operationalize privacy law by (i) generating use cases from mobile app descriptions, (ii) classifying them as compliant or non-compliant depending on whether consent requirements are satisfied, and (iii) modifying non-compliant use cases into compliant ones. In contrast to our work on generating Gherkin behavioural specifications, their work remains centred on classifying and repairing structured use cases for legal conformity.

Karpurapu et al. [136] generate BDD acceptance tests, finding that both zero-shot and few-shot prompting can produce high-quality scenarios, with few-shot prompts leading to better adherence to best practices. While their work focuses on optimizing prompt engineering for agile BDD practices, our study takes a different direction: using Gherkin specifications as a means to transform legal provisions.

Gokhan et al. [77] present a question-answering system for financial regulations and introduce a metric to assess the completeness of obligations addressed in the generated answers. Fuchs et al. [70] translate regulatory texts into formal logic. Abualhaija et al. [5] develop a Retrieval-Augmented Generation (RAG) pipeline that identifies relevant GDPR clauses and prompts an LLM to extract privacy obligations. Collectively, these strands of

work underscore the versatility of LLMs in legal compliance analysis, yet they do not systematically derive or evaluate behavioural specifications from legal texts through human-subject experiments. Our work addresses this gap by evaluating Gherkin specification generation from legal texts through a quasi-experimental study.

5.8.2 BDD Specification Quality Assurance

Ensuring high-quality user stories and BDD scenarios is challenging, particularly due to issues such as redundancy and ambiguity. To address these challenges, researchers have proposed quality frameworks and structured guidelines to support the writing and evaluation of user stories and BDD scenarios.

Early work by Heck et al. [110] proposes a quality verification framework tailored to agile requirements (e.g., feature requests and user stories), structured around three main criteria: completeness, uniformity, and conformance. This approach includes a checklist for evaluating agile requirements, with items such as a clear summary and description, rationale, relevant tags, unique identifiers, atomicity, and adherence to quality principles such as SMART [54] and INVEST [225]. Lucassen et al. [149,150] also propose the Quality User Story (QUS) framework for evaluating user-story quality across syntactic, semantic, and pragmatic dimensions. To operationalize the framework, they propose AQUSA (Automatic Quality User Story Artisan), a lightweight NLP-based tool that detects and highlights quality defects, such as ambiguity, lack of atomicity, and weak conceptual grounding.

Building on agile requirements quality work, Oliveira et al. [168,169] propose a checklist for BDD scenarios including uniqueness, integrity, essentiality, singularity, completeness, clarity, focus, and ubiquity. Wautelet et al. [230] in a subsequent study focuses on attributes such as uniqueness, essentiality, integrity, and singularity. Mapping studies [31] and suite-level principles by Binamungu et al. [30] extend this perspective to ensure the whole BDD suites remain coherent, adaptable, and understandable, whereas our study does not target the BDD feature suite.

Building on these studies, our work focuses on the automated generation of Gherkin specifications from legal provisions. Rather than introducing new evaluation metrics, we adapt quality attributes from the literature, *Relevance* (aligned with essentiality in [169]), *Completeness*, *Clarity*, and *Singularity*, as defined in Table 5.1, to evaluate the effectiveness of LLMs in generating Gherkin specifications. We also introduce *Time Savings* as a new criterion. Oliveira et al. [169] operationalize the attributes through checklist-based characteristics. Essential is mapped to characteristics such as: 1) (Concise) not too many details; not too many steps; no unnecessary lines. 2) (Small) not many steps. 3) (Steps) few and short steps; steps repetition. 4) (Additional Constructions) background; data tables; scenario outline. Complete is mapped to: 1) (Testable) follow the steps; completeness. 2) (Understandable) self-contained. 3) (Unambiguous) completeness. 4) (Additional Characteristics) complete. Clear is mapped to: (Language) technical jargon (Concise) clear (Unambiguous) vague statements; high granularity steps descriptions Singular is mapped to: 1) (Small) test one single thing. 2) (Testable) clear outcomes and verifications; clear

and simple Givens. 3) (Unambiguous) single clear intention; scenarios testing the same thing [169].

5.9 Data Availability

Our complete replication package is available on GitHub [104]. The package includes our dataset [105], complementary results from statistical analysis [106], and the implementation of additional algorithms, including statistical evaluation scripts [99].

5.10 Conclusion

In this chapter, we reported on a quasi-experimental study that evaluates the effectiveness of LLMs, specifically Claude 3.7 Sonnet [17] and Llama 3.3 70B Instruct [124], in automating the derivation of behavioural specifications from legal texts. With a focus on the domain of food-safety regulations, our work addresses a critical gap in requirements engineering: the labour-intensive and error-prone process of translating complex, legal provisions into structured, developer-friendly artifacts. By generating Gherkin specifications, a core element of Behaviour-Driven Development, we systematically examined whether LLMs can bridge this gap, producing outputs that are syntactically correct and semantically faithful to the source legal texts.

Our empirical results, based on 120 assessments from 10 human participants, show that LLM-generated Gherkin specifications are rated highly on all five quality criteria considered in our study: *Relevance*, *Clarity*, *Completeness*, *Singularity*, and *Time savings*. For *Relevance*, 75% of ratings achieved the top category "Fully relevant". *Clarity* stood out as a strength, with 90% top ratings, providing evidence for LLMs' ability to derive unambiguous Gherkin specifications from legal provisions. *Completeness* received 75% in the highest category, indicating broad coverage of functions and characteristics. *Singularity* received 81.7% in the highest category, reflecting strong alignment with single-purpose scenarios. *Time Savings* received 67.5% in the highest category, suggesting meaningful reductions in manual effort as perceived by participants. Wilcoxon Signed-Rank Test [41] tests between participants' ratings across all five quality criteria showed no reliable statistically significant differences. Furthermore, only three isolated cases of non-plausibility were identified, suggesting that nearly all the LLM-generated specifications are realistic.

A comparative analysis between Claude and Llama found no statistically significant differences across any quality criterion, as determined by Mann-Whitney U Tests. However, nuanced trends emerged: Llama slightly outperformed in *Clarity* (91.7% vs. 88.3% highest rating) and *Completeness* (76.7% vs. 73.3%), while Claude showed advantages in *Singularity* (85% vs. 78.3%), and *Time Savings* (71.7% vs. 63.3%).

Finally, qualitative feedback from participants provided additional insights, highlighting issues such as occasional hallucinations (e.g., fabricated details not present in the legal provisions), omitted clauses, and multi-intent scenarios.

For future work, we would like to investigate the derivation of behavioural specifications in other legal domains, such as privacy, data protection, and safety-critical laws, regulations, and standards. We would also like to explore how additional sources of information, such as organizational documents, process models, and system logs, could be integrated into the generation process to improve completeness and reduce omissions. Furthermore, we would like to consider strategies that make the derivation process more interactive, for example, by embedding human-in-the-loop feedback to iteratively refine and validate scenarios. Finally, we would like to broaden our study by involving professional practitioners as evaluators in order to better assess the usefulness of LLM-generated software development artifacts.

Chapter 6

Conclusion

This chapter concludes the thesis by synthesizing the findings from the three studies and showing how they collectively address the overarching *Thesis Research Questions* (TRQs). Beyond summarizing the individual contributions, the chapter reflects on their combined significance for regulatory compliance and requirements engineering. Finally, the chapter identifies important future work directions.

6.1 Addressing the Thesis Research Questions

The thesis set out to investigate how LLMs can enhance automation of critical tasks at the intersection of regulatory compliance and requirements engineering. To this end, three studies were conducted, each targeting a specific research question. The sections that follow revisit each TRQ in turn, linking it to the corresponding study and highlighting the key findings and contributions.

6.1.1 TRQ1: How can Grounded Theory and LLMs be used to identify and classify requirements-related provisions in underexplored regulatory domains, such as food safety regulations?

The first study presented in Chapter 3, *"An Empirical Study on LLM-based Classification of Requirements-related Provisions in Food-safety Regulations"* [107], bridges technology-agnostic food-safety regulations and monitoring software and systems by (i) deriving a software-relevant content model via a GT study and (ii) instantiating an automated classification pipeline with BERT and GPT variants to label provisions accordingly. The content model identifies three level-1 (L1) concepts–*Measurement*, *Data*, and *Time Constraint*–and level-2 (L2) refinements including *Temperature*, *Mass*, *Size*, *Label Data*, *Non-label Data*, *Colour*, *Firmness*, *Pathogen*, and *Water Content*.

Key findings and contributions:

- **Accurate automated classification with clear trade-offs.** Among fine-tuned models, GPT-4o achieved ≈89% *Precision* and ≈87% *Recall* on the Overall label, outperforming fine-tuned GPT-3.5 and BERT on *Precision* and matching/near-matching them on *Recall*; in contrast, few-shot GPT-4o increased *Recall* to ≈97% but reduced *Precision* to ≈65% (all differences are statistically tested).

- **Generalizes across jurisdictions.** Despite finetuning LLMs using Canadian regulations, the pipeline achieved comparable accuracy on selected U.S. FDA texts; *Recall* differences were not statistically significant, and GPT models showed higher *Precision* on FDA than on FSRG.

- **Only a minority of legal texts are software and system-relevant, motivating automation.** From 12,240 initial sentences across SFCR and FSRG, 9,105 were excluded as not directly related to food-safety system requirements, leaving 3,135 (≈26%) for further analysis.

- **Prevalence of software-relevant content in the dataset.** Within the labelled corpus, the Overall label (i.e., presence of at least one concept) appears in 54% of the fine-tuning set F and 56% of the test set T, which indicates that roughly half of the provisions contain software-relevant content.

- **Handling scarce concepts.** Several L2 concepts (*Colour*, *Firmness*, *Pathogen*, *Water Content*) are rare; LLMs struggled to classify these rare concepts, so the study complemented LLMs with keyword-based classification and reported separate results for these scarce labels.

This work provides the foundational framework for automating the identification of requirements-relevant provisions from regulations, addressing a critical gap in regulatory analysis in the food safety domain.

6.1.2 TRQ2: How does expanding LLM analysis to broader textual scopes (e.g., paragraphs) improve regulatory compliance automation in domains like data privacy regulations?

The second study in Chapter 4, *"Rethinking Legal Compliance Automation: Opportunities with Large Language Models"* [109], proposes shifting from sentence-level analysis to paragraph-level context and using generative LLMs for end-to-end compliance checking (with explanations) over GDPR-governed DPAs.

Key findings and contributions:

- **Paragraph context yields large accuracy gains.** Zero-shot results show mean *Accuracy* improvements when moving from sentences to paragraphs:
  GPT-3.5-turbo-0125: 30% → 63%;
  Mixtral-8x7B-Instruct-v0.1: 33% → 69%;
  GPT-4-0125-Preview: 41% → 81% (up to +40 points).

- **End-to-end decisions with justifications.** The approach elicits rule decisions and rationale from generative LLMs, addressing the justification gap in prior strategies.
- **Finer-grained assessments are feasible.** Unlike single-input classifiers that struggle to distinguish partial from full compliance, generative LLMs can articulate when rules are only partially met, to align better with legal nuances.
- **Benchmarking suggests sizable gains over a BERT baseline.** A re-implementation of a BERT-based approach (two prevalent rules) yielded *F-score* 67%; the lowest *F-score* observed by the proposed approach with GPT-4 exceeded 80%.
- **Efficiency and cost.** The pipeline averages approximately 0.7 s/paragraph (>5,000 paragraphs/hour). For a typical DPA, token usage translated to about $0.026 (GPT-3.5) and $0.27 (GPT-4) under January 2024 pricing.
- **Caveats and scope.** Justification interpretability was not yet evaluated; optimal context size beyond paragraphs remains an open question. The study's findings are still preliminary and only centred on DPAs, leaving other regulatory artifacts untested.

This study advances the state of compliance automation by introducing context-aware and end-to-end approaches that are more efficient and interpretable.

6.1.3 TRQ3: How can LLMs be leveraged to improve the automation of the derivation of behavioural specifications from regulatory texts?

The third study presented in Chapter 5, *"From Law to Gherkin: A Human-Centred Quasi-Experiment on the Quality of LLM-Generated Behavioural Specifications from Food-Safety Regulations"* [108], investigates how generative LLMs can translate complex legal texts into human- and machine-readable artifacts for software engineering by automatically deriving Gherkin specifications with a focus on food-safety provisions. Using Claude 3.7 Sonnet and Llama 3.3 70B Instruct, the study generated *60* specifications that were evaluated by *10* participants, yielding *120* assessments across five criteria: *Relevance*, *Clarity*, *Completeness*, *Singularity*, and *Time Savings*.

Key findings and contributions:

- **Rater consistency.** Pairwise comparisons between the two raters in each participant pair (Mann–Whitney U) showed no statistically significant differences for any criterion (all $p > 0.05$), with Vargha–Delaney $\hat{A}_{12}$ mostly negligible or small, indicating consistent perceptions across raters.
- **Quantitative results (combined across models).** Top-category rates were: *Relevance* **75%** "Fully relevant" (95% in the top two; *no* "Fully irrelevant"); *Clarity* **90%**

"Fully clear"; *Completeness* **75%** "Fully complete" (**94.2%** in the top two); *Singularity* **81.7%** "Fully singular"; *Time Savings* **67.5%** "Maximum time saved" (**91.7%** in the top two; *no* "No time saved"). Four "not plausible" ratings were recorded, corresponding to three unique specifications judged implausible.

- **Qualitative feedback.** Participants left **105** comments: 61% flagged issues (e.g., omissions, hallucinations/off-topic content, mixed-purpose scenarios) and 39% were approvals or refinement suggestions. By criterion, comments were distributed as: Time Savings 39, Relevance 30, Completeness 30, Singularity 22, Clarity 12.

- **Issue patterns.** Omissions (entire requirements, specific clauses, or details) mainly impacted *Completeness*; hallucinated or off-topic details reduced *Relevance*; conflated or split intents affected *Singularity* (and sometimes *Clarity*).

- **Per-model nuance (no statistically significant differences).** Llama and Claude performed comparably overall (no statistically significant differences; negligible effect sizes). Small trends: Llama slightly higher in *Clarity* (91.7% vs. 88.3%) and *Completeness* (76.7% vs. 73.3%); Claude higher in *Singularity* (85% vs. 78.3%); *Relevance* 75% top-tier for both; *Time Savings* favored Llama (71.7% vs. 63.3%).

6.1.4 Summary

Taken together, the three studies demonstrate how LLMs can be systematically leveraged across tasks such as:

1. Identifying and classifying requirements-related provisions in underexplored domains (food safety);

2. Assessing compliance at broader contextual levels with explanations (data privacy and policy); and

3. Deriving executable behavioural specifications from regulatory texts (food safety).

By addressing these stages in turn, the thesis establishes an end-to-end perspective that connects raw legal provisions to software artifacts and compliance assessments.

Beyond individual contributions, these studies collectively advance the state of research in two important ways. First, they demonstrate how regulatory compliance and requirements engineering can be jointly enhanced through LLM-based approaches that strike a balance between accuracy, interpretability, and efficiency. Second, they provide empirical evidence, ranging from quantitative benchmarks to human-subject evaluations, that demonstrates both the potential and the current limitations of LLMs in high-stakes, regulation-driven settings.

6.2 Future Work

Beyond the specific work discussed in this thesis, we outline seven broader research directions to guide future work at the intersection of regulatory compliance and RE.

- **Generalization across domains and jurisdictions.** Previous chapters identified the need to apply the ideas investigated in the proposed pipelines to new legal texts (e.g., additional food-safety bodies beyond North America, privacy DPAs beyond our current corpus) and to other regulated domains (e.g., finance) to better assess transferability and generalizability.

- **Adaptive context windows.** There is an opportunity to go beyond this thesis work (especially Chapter 4) by comparing fixed paragraph windows to *adaptive windows* that would expand according to the context, measuring the trade-off between decision accuracy and cost.

- **Specification improvement guided by lessons learned.** Using the error/lesson taxonomy from our human experiment (e.g., off-topic content, omissions, and mixed scenarios), a more iterative and corrective approach could involve an LLM-based *agent* that would produce revised Gherkin specifications that explicitly address the flagged issues, and then measure iteration-to-iteration gains over quality criteria.

- **Traceability from legal texts to requirements, models, tests, and runtime.** Building on the ideas investigated on requirements-relevant provision extraction from legal texts, paragraph-level compliance checking of regulatory artifacts with justifications, BDD specification generation from requirements-related provisions in legal texts, and recent work (e.g., by Kesari et al. [138] and Anmol and Breaux [200]), there are promising steps towards LLM-assisted compliance checking of legal rules. One important next step here would be to design pipelines that automatically *trace* extracted legal requirements to RE artifacts (including goals, process, and data models), to executable acceptance tests, and to runtime controls and audit logs.

- **Open benchmarks at the intersection of regulatory compliance and RE.** In an open-science spirit, it would be beneficial for the research community to build RE-related *datasets and benchmarks* (on food safety, privacy, health, finance, etc.) that are multi-jurisdictional, multilingual, and time-versioned (e.g., to capture regulation evolution and its impact on software and systems). These datasets would include feature-layered annotations, for example, deontic modalities (obligation, permission, and prohibition), domain-specific software-relevant labels, traceable cross-references, and gold-standard justifications. These would enable simpler, more extensive, and fairer comparisons between emerging approaches and technologies.

- **Faithful, auditable justifications aligned to evidence and formal artifacts.** In line with the idea investigated on the importance of justifications for compliance decisions and recent studies (e.g., by Amaral et al. [10] and Abualhaija et al. [5]), it

would be important to collaborate closely with legal experts to extract legal requirements, and automate ML/LLM solutions showing how expert-in-the-loop pipelines can scale legal analysis while keeping *faithfulness* to the source regulation. There is room here to investigate role-based workflows (with explicit responsibilities across legal experts, RE engineers, and testers), active learning, and weak supervision loops. Frequent missing elements include User Experience patterns and APIs for expert review, field studies on effort vs. risk reduction, and standards acceptable to auditors.

- **Regulation evolution analysis and real-time compliance impact.** Another research direction would be to extend current approaches on regulatory analysis, in line with the proposed studies in the thesis that build pipelines and processes involving legal texts, to take into account the impact of *changes* in regulations. In this context, it would be important to investigate how to incorporate deprecated provisions and new provisions into LLM- or other AI-based frameworks for regulatory analysis and compliance checking, in line with recent work (e.g., by Abualhaija et al. [4]) that analyzes changes across versions of regulations, to operationalize regulatory change over time, from initial parsing to runtime compliance checks. Areas to further explore include model update strategies, Retrieval Augmented Generation (RAG), and continual-learning approaches that ingest new or changed provisions without negative side-effects, and guardrails to prevent outdated rules from influencing decisions.

Appendices

Appendix A

Supplementary Material for Chapter 3

A.1 F1 Scores

F1-score plots are provided here (Figs. A.1 and A.2) for completeness. Given that *F1 scores* are a standard metric in classification tasks, we include them here in the appendix to ensure transparency and allow interested readers to cross-reference overall performance trends.

A.2 Original Provisions

The original provisions for examples provided (Fig. 3.1 in Section 3.1) are as follows:

– P1: Poultry Carcasses (other than air chilled) with weight between 1.8 Kg to 3.6 kg must be continuously chilled to reach less than 4°C in an additional time of 4 hours.

– P2: Grade Requirements for Canada C1 6. An ovine carcass graded Canada C1 must have

(a) the maturity characteristics set out in Schedule 1 – Maturity Characteristics for Lamb Carcasses to this Part;

(b) a minimum musculature score of 1 for each primal cut or a minimum average musculature score of less than 2.6;

(c) flank muscles that are pink to light red in colour; and

(d) a fat covering that

(i) is firm and white or slightly tinged with a reddish or amber colour, and

(ii) is less than 4 mm in thickness at the measurement site.

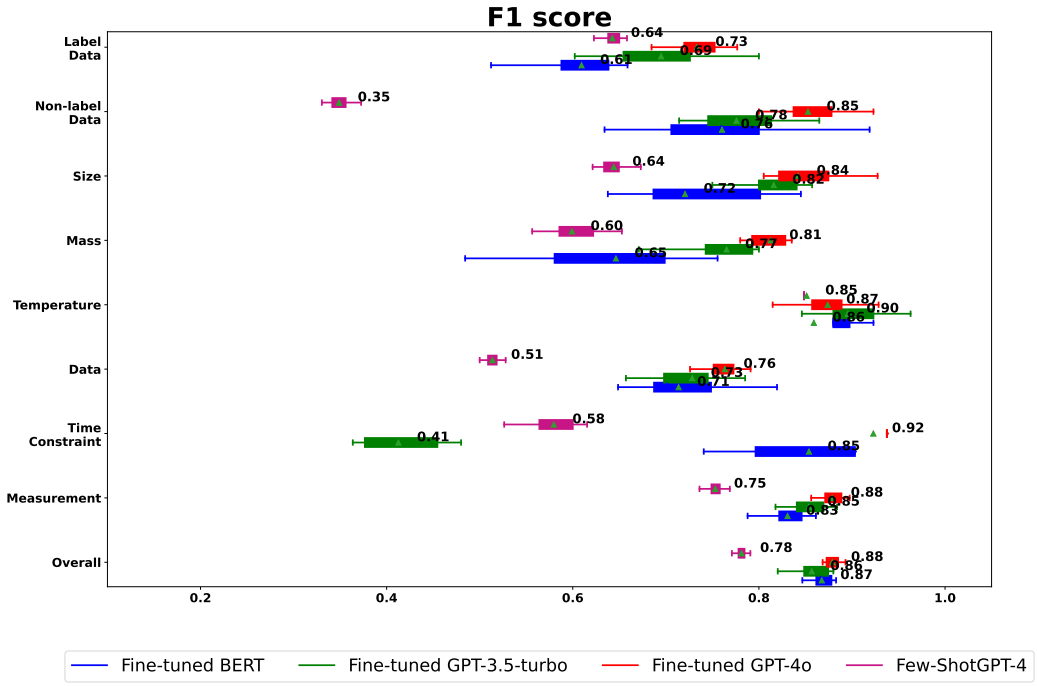


Figure A.1: *Precision* for fine-tuned BERT base, fine-tuned GPT-3.5-turbo, fine-tuned GPT-4o, and GPT-4o with few-shot learning (the x-axis represents the *F1 score* values).

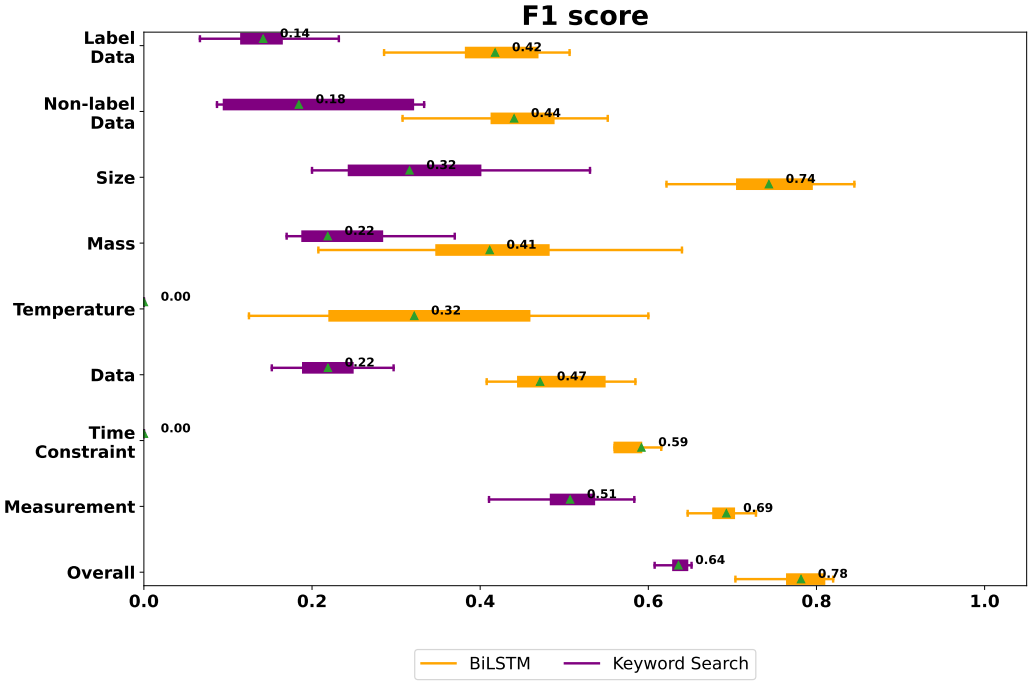


Figure A.2: *F1 score* for BiLSTM and Keyword Search (the x-axis represents the *F1 score* values).

– P3: 221 A label that is applied or attached to a consumer prepackaged food must bear on the principal display panel a declaration of net quantity of the food.

– P4: 90 (1) Any person who sends or conveys a food from one province to another, or who imports or exports it, any holder of a licence to slaughter a food animal, to manufacture, process, treat, preserve, grade, store, package or label a food or to store and handle an edible meat product in its imported condition and any person who grows or harvests fresh fruits or vegetables that are to be sent or conveyed from one province to another or exported must, if they provide the food to another person, prepare and keep documents that set out

(a) the common name of the food, a lot code or other unique identifier that enables the food to be traced and the name and principal place of business of the person by or for whom the food was manufactured, prepared, produced, stored, packaged or labelled;

(b) except if they provide the food to another person as a sale at retail, the date on which it was provided and the name and address of the person to whom it was provided;

(c) if they were provided the food by another person, the name and address of that person and the date on which it was provided; and

(d) the name of any food commodity that they incorporated into the food or from which they derived the food and, if they were provided the food commodity by another person, the name and address of that person and the date on which it was provided.

Remarks. We make two remarks regarding our modified provisions. First, they are intended for illustrative purposes and do not cover all the details, especially concerning P2 and P4. Our examples focus on conveying the core ideas rather than capturing every nuance and subtlety. Second, the actual text of the SFCR serves as a clear example of how legal language can complicate the interpretation of regulatory documents, further emphasizing the value of automated classification.

A.3 Grounded Theory Extracts

We have made our labelling document available, which includes colour-coded sheets, raised questions, notes, and relevant links (bit.ly/3G06g8R). Below, some parts of our preliminary iterations and some schema from our GT process are depicted, namely:

- List of FSRGs (Fig. A.3)
- Egg and processed egg (Fig. A.4)
- Meat products and food animals (Fig. A.5)
- SFCR

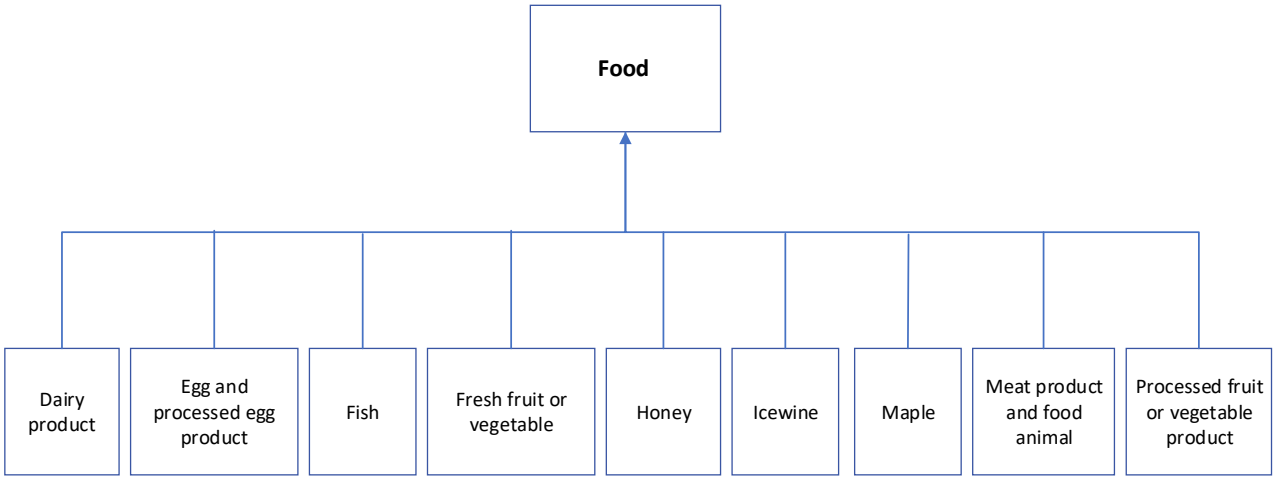


Figure A.3: Specific food categories in SFCR.

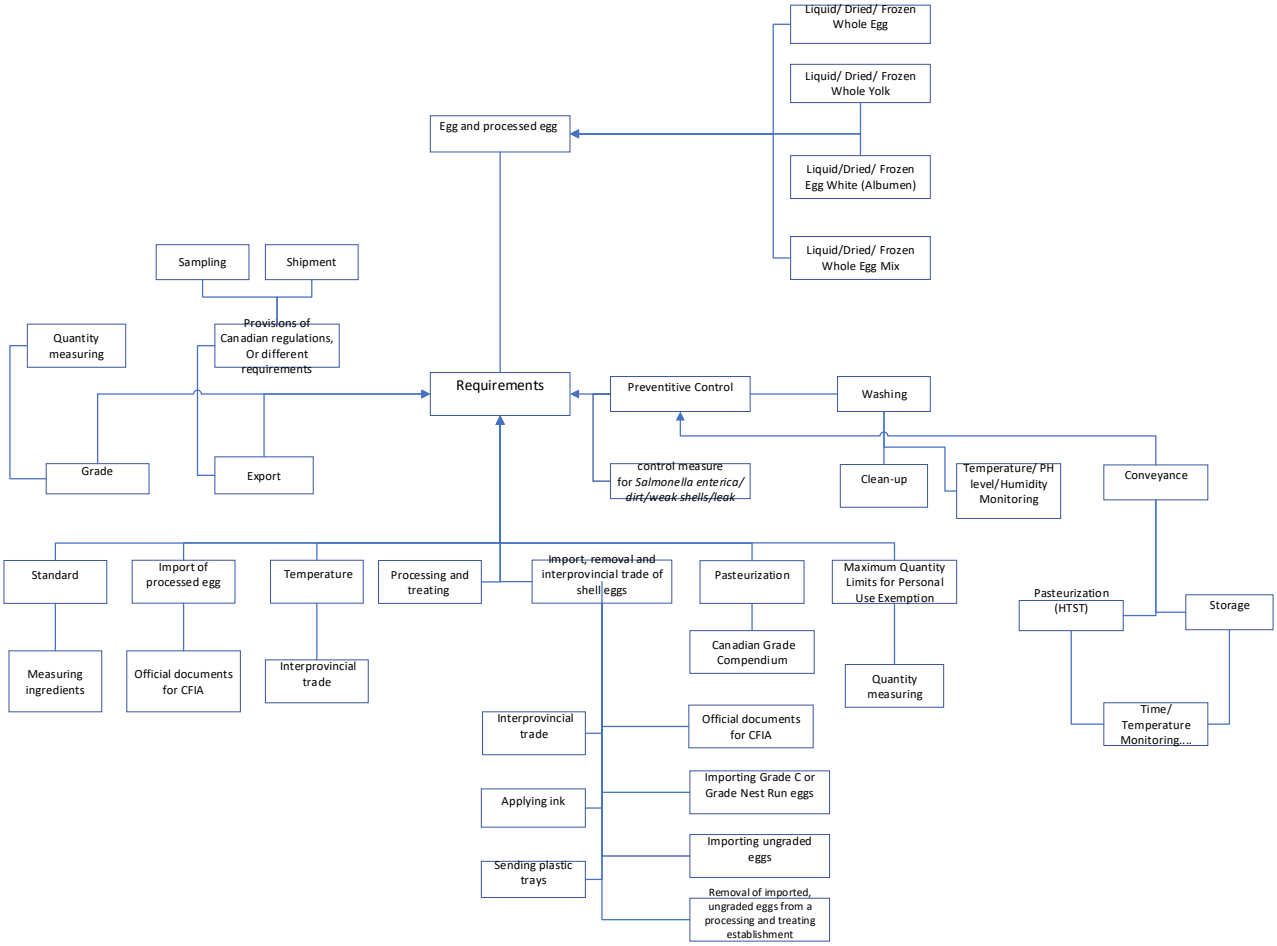


Figure A.4: Egg and processed egg food product categories and concepts.

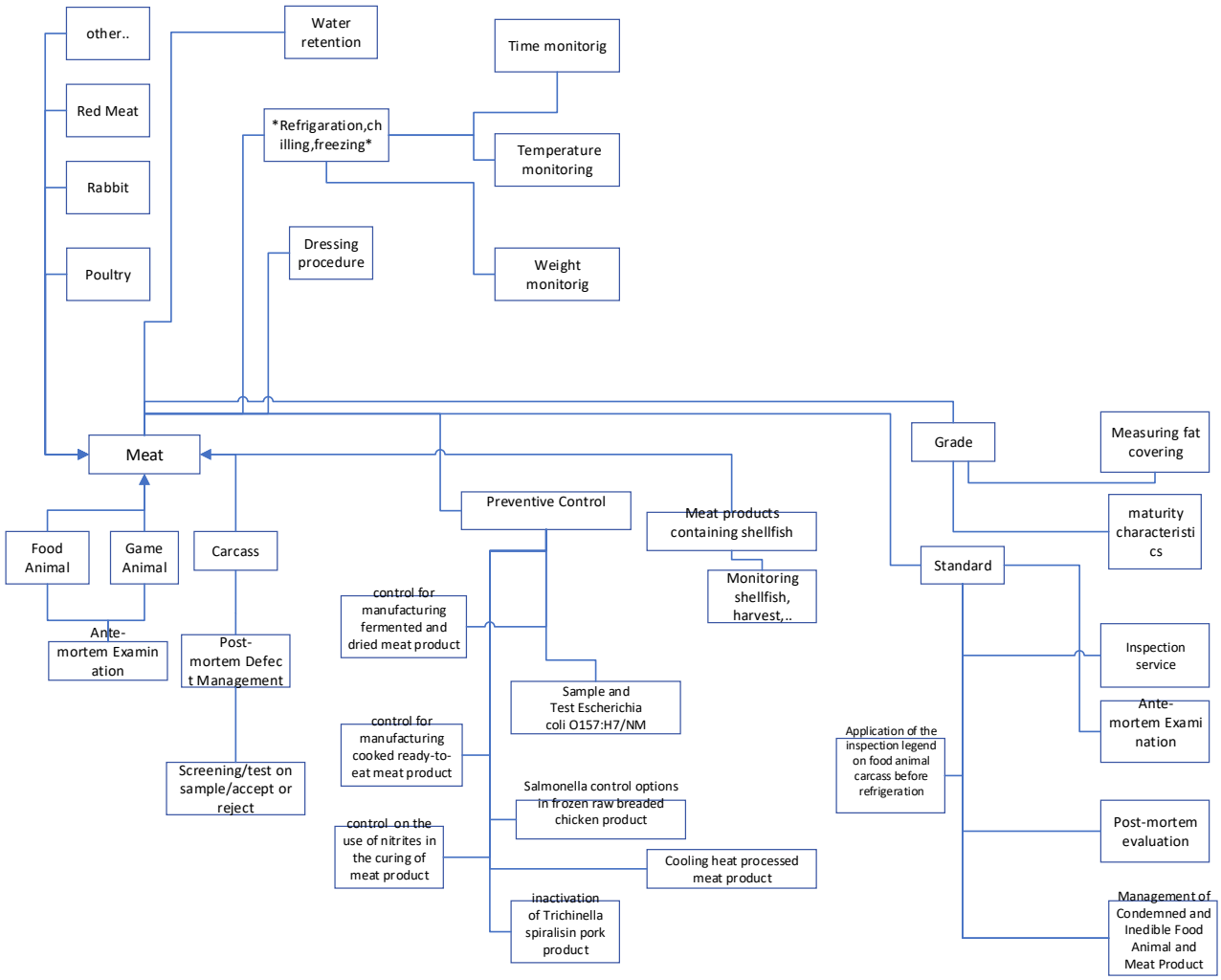
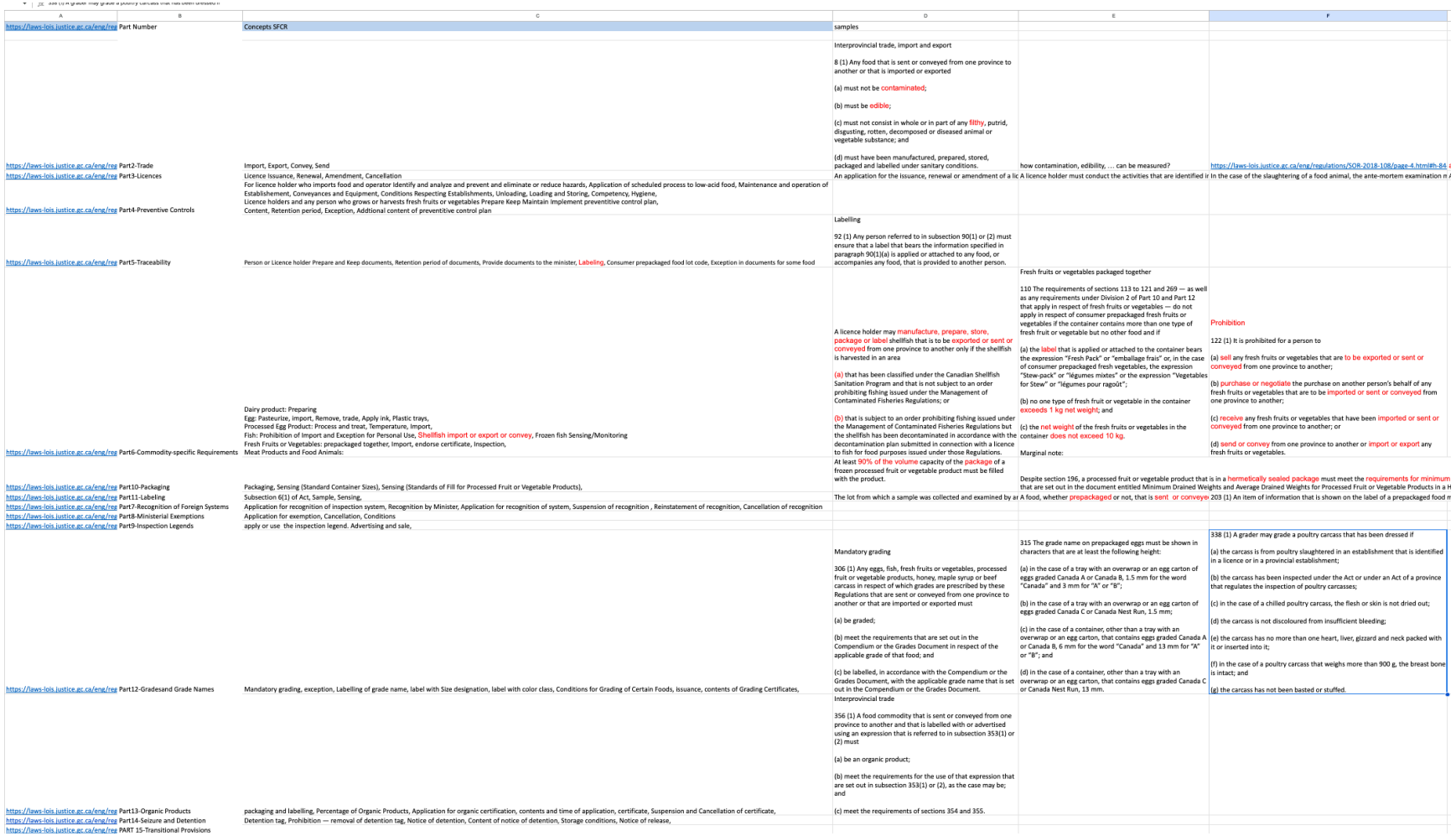


Figure A.5: Meat products and food animals categories and concepts.



A	B	C	D	E	F
https://laws-lois.justice.gc.ca/eng/reg	Part Number	Concepts SFCR	samples		
			Interprovincial trade, import and export 8 (1) Any food that is sent or conveyed from one province to another or that is imported or exported (a) must not be contaminated; (b) must be edible; (c) must not consist in whole or in part of any filthy, putrid, disgusting, rotten, decomposed or diseased animal or vegetable substance; and (d) must have been manufactured, prepared, stored,		
https://laws-lois.justice.gc.ca/eng/reg	Part2-Trade	Import, Export, Convey, Send	packaged and labelled under sanitary conditions.	how contamination, edibility, . . . can be measured?	https://laws-lois.justice.gc.ca/eng/regulations/SOR-2018-108/page-4.html#h-84
https://laws-lois.justice.gc.ca/eng/reg	Part3-Licences	Licence Issuance, Renewal, Amendment, Cancellation	An application for the issuance, renewal or amendment of a lic	A licence holder must conduct the activities that are identified ir	In the case of the slaughtering of a food animal, the ante-mortem examination n
https://laws-lois.justice.gc.ca/eng/reg	Part4-Preventive Controls	For licence holder who imports food and operator Identify and analyze and prevent and eliminate or reduce hazards, Application of scheduled process to low-acid food, Maintenance and operation of Establishement, Conveyances and Equipment, Conditions Respecting Establishments, Unloading, Loading and Storing, Competency, Hygiene, Licence holders and any person who grows or harvests fresh fruits or vegetables Prepare Keep Maintain Implement preventitive control plan, Content, Retention period, Exception, Addtional content of preventitive control plan			
https://laws-lois.justice.gc.ca/eng/reg	Part5-Traceability	Person or Licence holder Prepare and Keep documents, Retention period of documents, Provide documents to the minister, Labeling, Consumer prepackaged food lot code, Exception in documents for some food	Labelling 92 (1) Any person referred to in subsection 90(1) or (2) must ensure that a label that bears the information specified in paragraph 90(1)(a) is applied or attached to any food, or accompanies any food, that is provided to another person.		
https://laws-lois.justice.gc.ca/eng/reg	Part6-Commodity-specific Requirements	Dairy product: Preparing Egg: Pasteurize, import, Remove, trade, Apply ink, Plastic trays, Processed Egg Product: Process and treat, Temperature, Import, Fish: Prohibition of Import and Exception for Personal Use, Shellfish import or export or convey, Frozen fish Sensing/Monitoring Fresh Fruits or Vegetables: prepackaged together, Import, endorse certificate, Inspection, Meat Products and Food Animals:	A licence holder may manufacture, prepare, store, package or label shellfish that is to be exported or sent or conveyed from one province to another only if the shellfish is harvested in an area (a) that has been classified under the Canadian Shellfish Sanitation Program and that is not subject to an order prohibiting fishing issued under the Management of Contaminated Fisheries Regulations; or (b) that is subject to an order prohibiting fishing issued under the Management of Contaminated Fisheries Regulations but the shellfish has been decontaminated in accordance with the decontamination plan submitted in connection with a licence to fish for food purposes issued under those Regulations.	Fresh fruits or vegetables packaged together 110 The requirements of sections 113 to 121 and 269 — as well as any requirements under Division 2 of Part 10 and Part 12 that apply in respect of fresh fruits or vegetables — do not apply in respect of consumer prepackaged fresh fruits or vegetables if the container contains more than one type of fresh fruit or vegetable but no other food and if (a) the label that is applied or attached to the container bears the expression "Fresh Pack" or "emballage frais" or, in the case of consumer prepackaged fresh vegetables, the expression "Stew-pack" or "légumes mixtes" or the expression "Vegetables for Stew" or "légumes pour ragoût"; (b) no one type of fresh fruit or vegetable in the container exceeds 1 kg net weight; and (c) the net weight of the fresh fruits or vegetables in the container does not exceed 10 kg. Marginal note:	Prohibition 122 (1) It is prohibited for a person to (a) sell any fresh fruits or vegetables that are to be exported or sent or conveyed from one province to another; (b) purchase or negotiate the purchase on another person's behalf of any fresh fruits or vegetables that are to be imported or sent or conveyed from one province to another; (c) receive any fresh fruits or vegetables that have been imported or sent or conveyed from one province to another; or (d) send or convey from one province to another or import or export any fresh fruits or vegetables.
https://laws-lois.justice.gc.ca/eng/reg	Part10-Packaging	Packaging, Sensing (Standard Container Sizes), Sensing (Standards of Fill for Processed Fruit or Vegetable Products),	At least 90% of the volume capacity of the package of a frozen processed fruit or vegetable product must be filled with the product.	Despite section 196, a processed fruit or vegetable product that is in a hermetically sealed package must meet the requirements for minimum that are set out in the document entitled Minimum Drained Weights and Average Drained Weights for Processed Fruit or Vegetable Products in a H	
https://laws-lois.justice.gc.ca/eng/reg	Part11-Labeling	Subsection 6(1) of Act, Sample, Sensing,	The lot from which a sample was collected and examined by a	A food, whether prepackaged or not, that is sent or conveyed	203 (1) An item of information that is shown on the label of a prepackaged food m
https://laws-lois.justice.gc.ca/eng/reg	Part7-Recognition of Foreign Systems	Application for recognition of inspection system, Recognition by Minister, Application for recognition of system, Suspension of recognition , Reinstatement of recognition, Cancellation of recognition			
https://laws-lois.justice.gc.ca/eng/reg	Part8-Ministerial Exemptions	Application for exemption, Cancellation, Conditions			
https://laws-lois.justice.gc.ca/eng/reg	Part9-Inspection Legends	apply or use the inspection legend. Advertising and sale,			
https://laws-lois.justice.gc.ca/eng/reg	Part12-Gradesand Grade Names	Mandatory grading, exception, Labelling of grade name, label with Size designation, label with color class, Conditions for Grading of Certain Foods, issuance, contents of Grading Certificates,	Mandatory grading 306 (1) Any eggs, fish, fresh fruits or vegetables, processed fruit or vegetable products, honey, maple syrup or beef carcass in respect of which grades are prescribed by these Regulations that are sent or conveyed from one province to another or that are imported or exported must (a) be graded; (b) meet the requirements that are set out in the Compendium or the Grades Document in respect of the applicable grade of that food; and (c) be labelled, in accordance with the Compendium or the Grades Document, with the applicable grade name that is set out in the Compendium or the Grades Document.	315 The grade name on prepackaged eggs must be shown in characters that are at least the following height: (a) in the case of a tray with an overwrap or an egg carton of eggs graded Canada A or Canada B, 1.5 mm for the word "Canada" and 3 mm for "A" or "B"; (b) in the case of a tray with an overwrap or an egg carton of eggs graded Canada C or Canada Nest Run, 1.5 mm; (c) in the case of a container, other than a tray with an overwrap or an egg carton, that contains eggs graded Canada A or Canada B, 6 mm for the word "Canada" and 13 mm for "A" or "B"; and (d) in the case of a container, other than a tray with an overwrap or an egg carton, that contains eggs graded Canada C or Canada Nest Run, 13 mm.	338 (1) A grader may grade a poultry carcass that has been dressed if (a) the carcass is from poultry slaughtered in an establishment that is identified in a licence or in a provincial establishment; (b) the carcass has been inspected under the Act or under an Act of a province that regulates the inspection of poultry carcasses; (c) in the case of a chilled poultry carcass, the flesh or skin is not dried out; (d) the carcass is not discoloured from insufficient bleeding; (e) the carcass has no more than one heart, liver, gizzard and neck packed with it or inserted into it; (f) in the case of a poultry carcass that weighs more than 900 g, the breast bone is intact; and (g) the carcass has not been basted or stuffed.
https://laws-lois.justice.gc.ca/eng/reg	Part13-Organic Products	packaging and labelling, Percentage of Organic Products, Application for organic certification, contents and time of application, certificate, Suspension and Cancellation of certificate,	Interprovincial trade 356 (1) A food commodity that is sent or conveyed from one province to another and that is labelled with or advertised using an expression that is referred to in subsection 353(1) or (2) must (a) be an organic product; (b) meet the requirements for the use of that expression that are set out in subsection 353(1) or (2), as the case may be; and (c) meet the requirements of sections 354 and 355.		
https://laws-lois.justice.gc.ca/eng/reg	Part14-Seizure and Detention	Detention tag, Prohibition — removal of detention tag, Notice of detention, Content of notice of detention, Storage conditions, Notice of release,			
https://laws-lois.justice.gc.ca/eng/reg	PART 15-Transitional Provisions				

Figure A.6: SFCR categories per part.

A.4 Guidelines for Annotators

Guidelines for third-party annotators are as follows:

Instructions:

1. SFCR includes information on General food requirements and guidance and sets out many requirements that apply across food commodities, along with food-specific requirements. Therefore, SFCR sets the overall context and framework for food safety. Reviewing SFCR, a set of codes and categories is induced. However, various regulations elaborate additional and more specific requirements for specific food products. A variety of food products are covered by the Food-specific requirements and guidance (FSRG), including dairy products, egg and processed egg products, fish, fruits and vegetables, meat and food animals, honey, icewine and maple syrup. Information in (FSRG) related to specific foods is not all covered by SFCR solely. However, there are numerous FSRGs and we are interested in the ones which refine or complement the already derived codes from SFCR. The principals are being selective and achieving a high level of saturation. Therefore, based on the initial screening of documents, we decided to exclude the last category from further investigation, on the basis of being a niche area that is country-specific and unlikely to be generalized. Also, based on the intuition that handling meat products and food animals or egg and processed egg products are more involved than the rest, we limit the scope of open coding to these parts.

   Moreover, like all legal texts, FSRG and SFCR have several cross-references that include intertwining content with other documents, e.g., SFCR enabling acts, e.g., Safe Food for Canadians Act (SFCA), Food and Drug Regulations (FDR) and Food and Drug Act (FDA). Cross-references we saw in those FSRGs and SFCRs prompted our investigation of FDR, FDA, and SFCR enabling acts.

2. You should not follow cross-references as a subject of processing, although there are previous works that identify cross-references intent and the way to navigate them. Some provisions require compliance with some cited provisions, or there are exceptions to the previous provisions; in these cases, the information content of the statement should imply whether a label will be assigned to a statement. For instance, "217 A prepackaged food must have a label that meets the requirements of these Regulations applied or attached to it in the manner set out in these Regulations." You don't follow the content of "these regulations".

3. We have a classification where requirements can be of types of data, measurement, and time constraint.

4. According to our coding procedure, a unit of analysis for annotation purposes is the sentence level. These sentences are determined by the sentence-splitter and might not always be grammatically correct since the sentence-splitter (spacy) uses its dependency parser to divide texts into sentences. Moreover, you should look at context as in provisions with nested bullets, the preface may impact the sub-bullets. In other

words, in the case of interrelated itemized sentences, you should check the context in order not to miss any potential label due to only considering the genuine meaning of the itemized sentence. Thus, you should read the preface and decide whether the sub-bullets add new data, measurement, or time constraint.

5. As we went through the provisions of the regulation, we noticed some words that might imply vagueness, such as hazard, contamination, necessary, close to, sufficient, appropriate, adequate, distinct, discernible, as much as, promptly, quickly, nearly, immediately, clearly, etc. The important point here is that you need to investigate whether the sentence has any content which implies a label in spite of these vague words. For instance, "(3) The geographic origin of a food must, subject to the requirements of any other federal or provincial law, be shown in close proximity to the name and principal place of business of the person by or for whom the food was manufactured, processed or produced." In this example, although determining the proximity is vague, the "geographic origin must be shown on the food product" is a data definition requirement.

   "230 The declaration of net quantity that is shown on the label of a consumer prepackaged food must

   (a) be in distinct contrast to any other information or pictorial representation on the label; and

   (b) show the numerical quantity in boldface type."

   In this example, "the declaration of net quantity" must be shown, is the data definition, and "distinct contrast" is vague.

   "If such a word or expression is shown on the label, it must be shown in close proximity to the common name.", "208 Any information that a label is required by these Regulations to bear must be clearly and prominently shown and readily discernible and legible to the purchaser under the customary conditions of purchase and use." Leaving aside vague words like close proximity, clearly, prominently, and discernible, "any information" or "such word or expression" are too generic to be considered data definitions.

   "11 (1) Any food that is imported must have been manufactured, prepared, stored, packaged and labelled in a manner and under conditions that provide at least the same level of protection as that provided by sections 47 to 81." In this example, the same level of protection is not known unless you take a look at the cross-reference. For example, it might mean sanitizing hands or washing tankers. The statement does not have enough content to be considered as a measurement. As a result, it is impossible to determine the amount of "same level of protection" just by reading this statement.

   "(d) be equipped with instruments to control, indicate and record any parameters that are necessary to prevent contamination of the food;" In this example, we don't know all the necessary "parameters" to prevent contamination (it is too generic). There are a number of parameters that can be referred to, such as temperature,

time monitoring, space requirements, and sanitation rules, but not all may require measurement.

"Where conventional tank chilling is used, care must be taken to ensure that: sufficient overflow of water is provided to ensure the removal of extraneous matter prior to final icing." Although "sufficiency" is a vague term, it is clear that measuring the water level to reach a specific level is essential.

However, in these two examples: "133 A licence holder must provide a food animal with sufficient space to prevent the suffering of, injury to or death of the animal." or "134 A licence holder must provide a food animal with sufficient ventilation to prevent the suffering of, injury to or death of the animal." "sufficient space" and "ventilation" to prevent suffering or injury seem not to be measurable and actionable.

It appears that the three following requirements mention measuring temperature, humidity, PH, and pressure in order to reach a point according to the rules, ignoring the vagueness of the terms "appropriate level" and "adequate quantity".

(a) "65 (1) The temperature and humidity level in a facility or conveyance where a food is manufactured, prepared, stored, packaged or labelled or where a food animal is slaughtered must be maintained at levels that are appropriate for the food or the food animal, as the case may be, and for the activity being conducted."

(b) "71 (1) An establishment must be supplied, as appropriate for the food or the food animal that is intended to be slaughtered, as the case may be, and for the activity being conducted, with water that is adequate in quantity, temperature, PH and pressure to meet the needs of the establishment;"

(c) "(c) must be capable of maintaining the temperature and humidity at levels that are appropriate for the food and, if necessary to prevent contamination of the food, be equipped with instruments that control, indicate and record those levels."

"(4) The container of a consumer prepackaged food that is set out in column 1 of items 5 to 10 of Table 2 of Schedule 3 may have any volume capacity that is set out in Table 7 of that Schedule, in the case of metric containers, or Table 8 of that Schedule, in the case of imperial containers." Although you don't know what the values for "volume capacity" in the schedule are, but the statement explicitly elaborates on the measurement of this dimension, where the container may have the same volume capacity of the referred values.

"(2) The declaration of net quantity of a serving must be shown in accordance with the requirements of sections 231 and 233 to 237 respecting the declaration of net quantity of the food;" Data definition is present in this example, but the cited sections are not examined.

In some cases it cannot be determined what label will be assigned to the statement without reading the cited sections, e.g., "(3) Paragraphs (1)(b) and (c) do not apply

in respect of any processed fruit or vegetable product that does not meet the requirements of the Regulations with respect to grade and that is sent or conveyed from one province to another, if it is labelled with the words "Substandard" or "sous-régulier"."

"(2) Subsection (1) does not apply in respect of

(a) a food additive;

(b) a beverage that contains more than 0.5% absolute ethyl alcohol by volume; or

(c) a food that is set out in Schedule 1 and that

(i) is unprocessed and is intended to be manufactured, processed or treated for use as a grain, oil, pulse, sugar or beverage,

(ii) has a label applied or attached to it, or accompanying it, that bears the expression "For Further Preparation Only" or "pour conditionnement ultérieur seulement", and

(iii) is not a consumer prepackaged food."

In this example, bullet (b) has a measurement, but not the other bullets.

6. Generally, we do not classify human actions, such as imports, exports, sales, conveys, deliveries, grading, removal, rejections, etc., because we are not sure if they are automated by software systems. For instance: "201 A food, whether prepackaged or not, that is sent or conveyed from one province to another or that is imported or exported, and whose label bears a common name printed in boldface type, but not in italics, in the Standards of Identity Document must meet any standard that applies in respect of that common name." In this example, the label bears an expression, is not data definition and it is stated that some standards must be met.

   Additionally, there are cases where a statement contains more than one decision. In these cases, if a decision asks for a label, the statement must have it. "333 (1) Ungraded eggs that are received at an establishment where eggs are graded by a licence holder must be graded and labelled with the applicable grade name that is set out in the Compendium or if they do not meet the requirements in respect of any grade that is set out in these Regulations, they must be rejected." In this example, the food product must be graded and labelled with "the applicable grade name" or it must be rejected.

Labels:

1. Data: This label applies to statements of type data definition, data collection, or data retention.

2. Data definition: Characterizes the information i.e., the statement is labelled as data definition when it defines information. It defines what data, the labels must bear. In most cases where there are phrases e.g., "must be labeled/graded with", "label must bear", "must be shown on", "label/grade must be attached/applied", "are required to be labelled with", "must be shown as","shall be located on" they elaborate on the

expressions, mark or stamp that must or may be shown on the label, grade or on the container of a food product. A few examples are listed below:

"254 The label of prepackaged eggs that are graded in accordance with these Regulations must bear

(a) a declaration of net quantity; and

(b) in the case of eggs that are pasteurized in the shell, the words "Pasteurized" and "pasteurisé", as well as the expressions "Graded Canada A Before Pasteurization" and "classifié Canada A avant pasteurisation" or the expressions "Graded Grade A Before Pasteurization" and "classifié catégorie A avant pasteurisation", as the case may be."

"(3) The relative firmness of the cheese must be identified by the following expressions:

(a) "Soft White Cheese" and "fromage à pâte fraîche" or "fromage frais", if it has a moisture on fat-free basis content of 80% or more;

(b) "Soft Cheese" and "fromage à pâte molle", if it has a moisture on fat-free basis content of more than 67% but less than 80%;"

"321 Fresh fruits or vegetables that are sent or conveyed from one province to another or imported must be labelled with the applicable size designation that is set out in the Compendium, if any. The size designation must

(a) be shown in close proximity to the grade name;

(b) in the case of prepackaged fresh fruits or vegetables, other than consumer prepackaged fresh fruits or vegetables,

(i) if their container is a reusable plastic container, be shown in characters that are at least 1.6 mm in height, or

(ii) if their container is not a reusable plastic container, be shown in characters of at least the minimum character height that is set out in paragraph 320(1)(b) for the grade name"

As an example, food products must be labeled with the relevant size designation, and that is a data definition since we know what size designation means. However, in the following example, the "descriptive term" doesn't define any data since we don't know what information it characterizes. "(3) The descriptive term referred to in paragraph (1)(e) must be shown in close proximity to the common name and in characters that are at least the height that is the greater of

(a) one-half the height of the characters in which the common name is shown, and

(b) 1.6 mm."

"233 (1) The metric units that must be shown in a declaration of net quantity of a consumer prepackaged food must be in

(a) millilitres, if the net volume of the food is less than 1 000 mL;

(b) litres, if the net volume of the food is 1 000 mL or more;

(c) grams, if the net weight of the food is less than 1 000 g; and

(d) kilograms, if the net weight of the food is 1 000 g or more."

This example shows that the preface is used in conjunction with the bullet (a) to define the data, where the metric units must be shown in milliliters, but other bullets are simply different formats of this data. Also, in the example below, the preface, along with bullet (a), defines quantity, while bullet (b) refers to a different format (decimal system) of the data. "235 If the declaration of net quantity of a consumer prepackaged food is shown in metric units and the quantity is less than one metric unit, the quantity must be shown

(a) in words; or

(b) in the decimal system, with a zero preceding the decimal point."

"231 The declaration of net quantity of a consumer prepackaged food must be shown

(a) in the case of a consumer prepackaged food that is listed in the document entitled Units of Measurement for the Net Quantity Declaration of Certain Foods, prepared by the Agency and published on its website, as amended from time to time, by volume, weight or numerical count in accordance with that document; or

(b) in the case of a consumer prepackaged food that is not listed in the document referred to in paragraph (a),

  (i) if the food is liquid, gas or viscous, by food volume, or, if the food is solid, by weight, or

  (ii) if the established trade practice in respect of the food is to show its net quantity in a manner that is different than what is required by subparagraph (i), in accordance with that established trade practice."

In this example, the declaration of net quantity must be shown by volume, weight, numerical count, or in accordance with some cross-references. Data definition is the first part of defining net quantity; however, we do not consider the following bullets to be data definitions.

"(2) If prepackaged fresh fruits or vegetables that are labelled in accordance with this Part are placed inside of a second container and the resulting product is prepackaged fresh fruits or vegetables, other than consumer prepackaged fresh fruits or vegetables, the second container is not required to be labelled with the information referred to in subsection (1) if that information is readily discernible and legible without having to open the second container and that information is not obscured by the container."

In this example, there is a cross-reference needed in order to understand what the "information" is. More importantly, the compliance here, "is not required to be labelled with" is permission, not a requirement.

In requirements such as the two mentioned below, data is not explicitly defined. To define data, we cannot use generic phrases, such as "that information" or "the statement".

"(2) That information must be shown in characters that are in the case of a tray with an overwrap or an egg carton, on the top or side of the tray or egg carton, at least 1.5 mm in height;"

"(2) The statement must be shown on or in close proximity to the pictorial representation, if the representation is shown on the principal display panel;"

3. Data collection: The statement is labelled as Data collection when it needs a document to be recorded. For instance:

   "(2) If a complaint is received, the operator must implement the procedure and prepare a document that sets out the details of the complaint, the results of the investigation and the actions taken based on those results and keep it for two years after the day on which the actions are completed."

   "The grading certificate must be signed by the grader and include the following information:

   (a) the name and address of the producer;

   (b) the name of any person who is acting on behalf of the producer;"

   "98 (1) Despite subsection 306(1), a licence holder may import ungraded eggs if they before the import, notify the Minister in writing of the quantity of ungraded eggs that are intended to be imported, the date of the import and the name of the licence holder and address of the establishment referred to in paragraph (c);"

   "(2) Any person who sells a food at retail, other than a restaurant or other similar enterprise that sells the food as a meal or snack, must prepare and keep documents that include the information specified in paragraphs (1)(a), (c) and (d)."

4. Data Retention: A statement is labeled data retention when a document needs to be maintained, such as, "(3) The documents referred to in subsections (1) and (2) must be kept for two years after the day on which the food was provided to another person or sold at retail, and must be accessible in Canada."

5. Sensing/Measurement: Measurements is the association of numbers with physical quantities, characteristics and phenomena such as temperature, dimension measures, weight, numerical count, color, firmness, PH level, fat content, water level, bacteria concentration, test, pressure, and humidity. You need to specify which measurement types you found. If a new measurement type is found in a statement, please wrtie it in the column named "others".

   "The thawing procedures must prevent net gain in weight over the frozen weight, when poultry products are thawed for repackaging." To prevent net weight gain in a specific phase, someone must assess and keep track of it.

"When water (including ice) is used for chilling and comes in contact with carcasses, carcass parts and giblets during chilling process, the water retention must be assessed." The requirement explicitly asks for an assessment of water retention.

"The poultry product will be considered "frozen" and labelled accordingly when freezing becomes more extensive i.e. deeper than 4mm." As an example, the product will be labelled based on its freezing depth.

"188 (1) The container of a consumer prepackaged food that is set out in column 1 of Table 1 of Schedule 3 must be of a size that corresponds to a net quantity by weight or by volume that is set out in column 2 or 3." It is necessary to measure the food product in this instance since its net quantity must match the container size.

"(2) Subject to subsection (3), if a lot contains the number of units set out in column 1 of Part 1 of Schedule 5, the inspector must collect from the lot at least the number of units set out in column 2 and the units collected constitute the sample referred to in subsection (1)." This example measures the number of units.

"(5) In the case of an inspection of a consumer prepackaged food that consists of a liquid, the net quantity of the food must be determined on the basis of the assumption that the liquid is at a temperature of 20°C." This example implies that the net quantity must be adjusted and selected based on liquid temperature constraint; therefore, the liquid temperature must be measured as a condition.

"(6) In the case of an inspection of a consumer prepackaged food that consists of a frozen liquid food and that is normally sold and consumed in a frozen state, the net quantity of the food must be determined when the food is in a frozen state.". When the food is in a particular state, the net quantity must be measured and determined.

"(3) For the purposes of subsection 6(1) of the Act, labelling a consumer prepackaged food with a declaration of net quantity does not constitute labelling a food in a manner that is false, misleading or deceptive if (b) the actual net quantity of the food is, subject to the tolerance provided under subsection (5), not less than its declared net quantity." One can determine the falseness, misleadingness, and deception of the labelling by measuring the net quantity value.

"343 The percentage of the contents of a multi-ingredient food commodity that are organic products must be determined in accordance with CAN/CGSB-32.310." In this example, "The percentage of the contents of a multi-ingredient food commodity" must be measured and determined.

"Product temperature requirements must be met at the time of shipping.", "If the timeframe has elapsed, the FSIS inspector will be required to monitor the temperature of the load to verify compliance, as per CFR 590.950(b).", and "48 (1) If a low-acid food is in a hermetically sealed package, an operator must apply the scheduled process referred to in subparagraph (3)(a)(i) and, if batch thermal treatment is applied, must use a temperature-sensitive indicator that visually indicates that the package has been thermally treated." These requirements discuss temperature monitoring that requires measurements.

"The licence holder must submit to the veterinary inspector a written protocol for each all poultry products which will be crust frozen outlining the time period and location for the equilibrium of internal and external product temperatures such that an internal temperature is between 4°C and -2°C.", "Use of the term "air chilled" or similar phrases must be restricted to carcasses or portions when licence holder can demonstrate through a written PCP program and validation data that there is no net increase in the weight of the carcasses as a result of post evisceration washing, chilling and drainage as per "Poultry Retained Water Control Program"." As these two examples require measurement requirements to be written, they are also data collection requirements.

You should note that just detecting the measures or the units of measurements e.g., °C , mm, cm2, g/kg, mL/L, pink, white, red and so forth is not enough. For instance: "272 (1) The label of a prepackaged processed fruit or vegetable product must bear (f) the total percentage of sweetening ingredients added, if any, in the case of frozen fruits packaged in sugar, invert sugar, dextrose or glucose in dry form;" In this example, the data definition on the label may contain some metric units.

"(2) Despite paragraphs (1)(a) and (c), 500 mL may be shown as 0.5 L and 500 g may be shown as 0.5 kg." and "(3) That information, other than the declaration of net quantity, may be shown in characters that are at least 0.8 mm in height if (a) the information that a label is required by Division 2 to bear is shown on the principal display panel" These statements outline the font and format of labels, which we consider as a convention.

6. Time constraint: Temporal restrictions or conditions. In this context, we have observed two main types of time constraints: interval/deadline and periodicity. Deadline: a point in time by which something must be done. Periodicity: the tendency, quality, or fact of recurring at regular intervals.

   In the following samples, you'll find examples of events that must or may occur in a fixed period of time or periodically.

   Interval/deadline: "91 (1) Any person who has received a request from the Minister for a document referred to in section 90, or any part of such a document, must provide it to the Minister

   (a) within 24 hours after receipt of the request, or within,

      (i) any shorter period that is specified by the Minister, if the Minister believes that it is necessary in order to identify or respond to a risk of injury to human health associated with a food commodity, or

      (ii) any longer period that is specified by the Minister, if the Minister believes that the document is not necessary for a recall that is or may be ordered under subsection 19(1) of the Canadian Food Inspection Agency Act; and

   (b) if provided electronically, in a single file and in plain text that is capable of being imported into and manipulated by standard commercial software."

In this example, bullet (a) has specifically mentioned the deadline as "24 hours after receipt of the request".

"(3) The documents referred to in subsections (1) and (2) must be kept for two years after the day on which the food was provided to another person or sold at retail, and must be accessible in Canada."

"138 (1) Within 24 hours before the slaughter of a food animal and in accordance with the document entitled Ante-mortem Examination and Presentation Procedures for Food Animals, prepared by the Agency and published on its website, as amended from time to time, a licence holder must conduct an ante-mortem examination of the food animal or of a sample from the shipment that the food animal is part of, which must include, in the case an equine or a bird other than one that is a game animal or an ostrich, a rhea or an emu, the examination of the documents referred to in subsection 165(1)."

"11(2) If a complaint is received, the operator must implement the procedure and prepare a document that sets out the details of the complaint, the results of the investigation and the actions taken based on those results and keep it for two years after the day on which the actions are completed."

"The process must be monitored, by a designated plant employee, a minimum of every 2 hours for crust disappearance and internal product temperature."

Periodicity:

"(2) The operator must, at least once every 12 months,

(a) conduct a recall simulation, based on the recall procedure;

(b) prepare a document that sets out the details of how the recall simulation was conducted and the results of the simulation, and keep that document for two years after the day on which the recall simulation is completed."

In this example, the first sentence (preface in combination with bullet (a)) mentions "every 12 months" which is consistent with periodicity time constraint. Keeping records for a period of time is required by bullet (b).

However, you can see samples that include time (even its metric unit) but does not imply time constraint, such as, "The delayed evisceration time must be incorporated into total chilling time for the affected product.", "The license holder can validate alternate delayed evisceration time (other than 30 minutes).", "When developing the monitoring procedure following must be followed: Select time for examination." or "225 The label of a prepackaged food must be applied or attached in such a manner that the label is still applied or attached at the time it is sold."

7. Convention: There are a number of requirements for the legibility/usability of labels. Most commonly, these include standards for the font (dimension measure) and format (colors, metric units, languages, and order) of the expressions/characters/-marks/information. We don't have control over measuring or deciding them, for example:

"(2) That information must be shown in characters that are

(a) in the case of a tray with an overwrap or an egg carton, on the top or side of the tray or egg carton, at least 1.5 mm in height; and

(b) in the case of a container other than a tray with an overwrap or an egg carton, at least 6 mm in height."

"(3) The geographic origin of a food must, subject to the requirements of any other federal or provincial law, be shown

(a) in close proximity to the name and principal place of business of the person by or for whom the food was manufactured, processed or produced; and

(b) in characters of at least the same height as those in which the information referred to in paragraph (a) is shown."

"(2) That information must be shown in characters that are

(a) in the case of a tray with an overwrap or an egg carton, on the top or side of the tray or egg carton, at least 1.5 mm in height; and

(b) in the case of a container other than a tray with an overwrap or an egg carton, immediately below the common name, at least 13 mm in height."

In this example, height (generally format) is a given, and it is not something we have control over.

Appendix B

Supplementary Material for Chapter 4

Here is the list of top the 25 most populated rules borrowed from Amaral et al. [10]:

R1: The DPA shall contain at least one controller's identity and contact details. (LL)

R2: The DPA shall contain at least one processor's identity and contact details. (LL)

R3: The DPA shall contain the duration of the processing. (Art. 28(3))

R4: The DPA shall contain the nature and purpose of the processing. (Art. 28(3))

R5: The DPA shall contain the types of personal data. (Art. 28(3))

R6: The DPA shall contain the types of personal data. (Art. 28(3))

R7: The organizational and technical measures to ensure a level of security can include:

(a) pseudonymisation and encryption of personal data,

(b) ensure confidentiality, integrity, availability and resilience of processing systems and services,

(c) restore the availability and access to personal data in a timely manner in the event of a physical or technical incident, and

(d) regularly testing, assessing and evaluating the effectiveness of technical and organisational measures for ensuring the security of the processing. (Art. 32(1))

R8: The notification of personal data breach shall at least include

(a) the nature of personal data breach;

(b) the name and contact details of the data protection officer;

(c) the consequences of the breach;

(d) the measures taken or proposed to mitigate its effects. (Art. 33(3))

R10: The processor shall not engage a sub-processor without a prior specific or general written authorization of the controller. (Art. 28(2))

R11: In case of general written authorization, the processor shall inform the controller of any intended changes concerning the addition or replacement of sub-processors. (Art. 28(2))

R12: The processor shall process personal data on documented instructions from the controller. (Art. 28(3a))

R15: The processor shall ensure that persons authorized to process personal data have committed themselves to confidentiality or are under an appropriate statutory obligation of confidentiality. (Art. 28(3b))

R16: The processor shall take all measures required pursuant to Article 32 or to ensure the security of processing. (Art. 28(3c))

R22: R22 - The processor shall assist the controller in ensuring compliance with the obligations pursuant to data protection impact assessment (DPIA). (Art. 28(3f), Art.35)

R23: The processor shall return or delete all personal data to the controller after the end of the provision of services relating to processing. (Art. 28(3g))

R24: The processor shall immediately inform the controller if an instruction infringes the GDPR or other data protection provisions. (Art. 28(3h))

R25 The processor shall make available to the controller information necessary to demonstrate compliance with the obligations Article 28 in GDPR. (Art. 28(3h))

R26 The processor shall allow for and contribute to audits, including inspections, conducted by the controller or another auditor mandated by the controller. (Art. 28(3h))

R27 The processor shall impose the same obligations referred to in Article 28(3) in GDPR on the engaged sub-processors by way of contract or other legal act under Union or Member State law. (Art. 28(4))

R28 The processor shall remain fully liable to the controller for the performance of sub-processor's obligations. (Art. 28(4))

R30 The processor shall not transfer personal data to a third country or international organization without a prior specific or general authorization of the controller. (Art. 28(3a))

R34 The processor shall notify the controller without undue delay after becoming aware of a personal data breach. (Art. 33(2))

R36 In case of general written authorization, the controller shall have the right to object to changes concerning the addition or replacement of sub-processors, after having been informed of such intended changes by the processor. (Art. 28(2))

R38 The controller shall have the right to terminate the DPA in certain cases. (LL)

R39 The controller shall, no later than 72 hours after having become aware of it, notify the personal data breach to the supervisory authority. (Art. 33(1))

Appendix C

Supplementary Material for Chapter 5

C.1 Content of Interview Protocol

Introduction: You are invited to participate in evaluating Gherkin specifications generated by a Large Language Model (LLM) (ChatGPT) as part of a research study aimed at improving software artifacts related to food safety monitoring systems. Your feedback is crucial to understanding the usability and accuracy of these AI-generated outputs.

Objective: Your task is to review the Gherkin specifications generated by (LLM) based on the regulatory provisions provided. We will use your review results to identify the strengths and weaknesses of LLMs in automatically generating Gherkin specifications. You are asked to provide a rating for each criterion per specification and provide feedback on identified issues to help us understand problems or room for improvement.

Background: The regulatory provisions are extracted from food-safety regulations. The provisions have been colour-coded to help you more easily see the food-safety concepts that they relate to. Use this information to guide your review and ensure the specifications reflect the intended concepts accurately. Gherkin specifications are expected to follow the **Given-When-Then** format:

1. Given describes the context or initial state of the system.
2. When specifies the event or action that triggers the behaviour.
3. Then defines the expected outcome or result of that action.

Evaluation Criteria: Use the following criteria to assess the Gherkin specifications:

1. Relevance: Specification matches the intended system behaviour as described in the regulatory text while focusing on what is directly pertinent.
   Key Points: Only pertinent details for the specified behaviour are presented, i.e., nothing is said that is not relevant.
   Scale:

(a) All irrelevant: The output does not address the requirement at all. The output is entirely off-topic or unnecessary.

(b) Mostly irrelevant: A small portion of the output addresses the requirement, while most content is off-topic or unnecessary.

(c) Somewhat relevant: A substantial portion of the output relates to the requirement, but an equally substantial portion remains off-topic or unnecessary.

(d) Mostly relevant: Most of the output addresses the requirement, with only a small portion being off-topic or unnecessary.

(e) All relevant: All content in the output addresses the requirement, with no off-topic or unnecessary information.

2. Completeness: Specification covers all intended functionality.

   Key Points: All functionalities implied by the regulatory requirement are included without omissions.

   Scale:

   (a) Fully incomplete: No required information is included. The specification omits all relevant information from the requirement.

   (b) Mostly incomplete: Only a small fraction of the required information is included, while most of the relevant information is missing.

   (c) Partially complete: A substantial portion of the required information is included, but an equally substantial portion is missing.

   (d) Mostly complete: Most of the required information is included, with only a small portion of relevant information missing.

   (e) Fully complete: All required information is included. The specification omits no relevant information.

3. Clarity: Specification is written in a clear and unambiguous manner.

   Key Points: 1)Specifications can be uniquely interpreted, i.e., not being vague or misleading. 2) Technical jargon commensurate with the need for specification.

   Scale:

   (a) Completely unclear: No part of the specification is understandable. The content is highly vague, ambiguous, or difficult to interpret.

   (b) Mostly unclear: Only a small portion of the specification is understandable, while most parts are vague, ambiguous, or difficult to interpret.

   (c) Somewhat clear: A substantial portion of the specification is understandable, but an equally substantial portion is vague, ambiguous, or difficult to interpret.

   (d) Mostly clear: Most parts of the specification are easy to understand, with only a small portion being vague, ambiguous, or difficult to interpret.

   (e) Completely clear: All parts of the specification are easy to understand, with no vague, ambiguous, or difficult-to-interpret content.

4. Integrity (later removed): Specification follows the BDD structure of Given-When-Then, i.e., Each scenario has a non-empty context, event, and outcome.

   Key Points: If the regulatory requirement is not written with a specific event or lacks context, the specification might not include valid steps. 2) Context might be included in the background rather than scenarios.

   (a) No adherence: The specification does not follow the BDD structure at all.
   (b) Minimal adherence: The specification follows the BDD structure in a few scenarios, while it does not follow it in most scenarios.
   (c) Somewhat adherent: The specification follows the BDD structure in a substantial number of scenarios, but it does not follow it in a substantial number of scenarios.
   (d) High adherence: The specification follows the BDD structure in most scenarios, while it does not follow it in a few scenarios.
   (e) Fully adherent: The specification strictly follows BDD structure in all scenarios.

5. Singularity: Each scenario focuses on a single purpose i.e., context, event, and outcome are provided to describe that purpose.

   Key Points: 1) Specification avoids mixing multiple purposes within a single scenario. 2) Specification splits scenarios when multiple functions or events serve distinct purposes to improve testability and maintainability.

   Scale:

   (a) Completely mixed: The output combines multiple purposes in each scenario, making testing and maintenance difficult.
   (b) Mostly mixed: The output separates purposes in a few scenarios but still combines multiple purposes in most, making testing and maintainability difficult.
   (c) Somewhat singular: The output features a substantial number of scenarios with a single purpose, but an equally substantial number that combine multiple purposes.
   (d) Mostly singular: The output ensures most scenarios focus on a single purpose, but a few still combine multiple purposes, making testing and maintainability somewhat difficult.
   (e) Completely singular: The output ensures each scenario has a single purpose, making testing and maintenance easy.

6. Time Savings: The extent to which the specification reduces the time a participant would otherwise spend creating the specification from scratch.

   Key Points: Reflects how much of the specification can be reused or adapted instead of fully rewritten.

   (a) Completely unhelpful (no time saved) Does not reduce the workload at all. The specification must be entirely rewritten from scratch.

(b) Mostly unhelpful (little time saved) Only a small fraction of the specification can be reused. Most parts require to be rewritten from scratch.

(c) Somewhat helpful (moderate time saved) A substantial portion of the specification can be reused, but an equally substantial portion still needs to be rewritten from scratch.

(d) Mostly helpful (significant time saved) Most of the specification can be reused. Only a small portion that needs to be rewritten from scratch.

(e) Completely helpful (maximum time saved) Completely eliminates manual work. The specification can be adopted virtually as-is.

To do:

1. Review best practices in writing BDD
2. Read the regulatory requirements carefully:

   a) Understand the intent and scope of the input.
3. Review each specification against the criteria:

   a) Use the provided criteria (Relevance, Completeness, Clarity, and Singularity, Time Savings) and the scale per criterion.
4. Provide constructive feedback:

   a) Be precise about issues per criterion e.g.,‘ ambiguous or unclear terms‘, ‘specific functionality not being covered‘, etc. to justify your ratings and present areas for improvement in your feedback.
5. Write your own version of the specification if you think your version is more plausible based on common sense.

   a) If you detect major issues with the given specification, it might be easier for you to rewrite it as an alternative to providing feedback (you still need to provide ratings for the given specification, though).

   b) Even if you strongly agree with the definition of criteria for the given specification, please write down your observations when you notice something interesting or room for improvement. For instance, If the specification is ‘all relevant‘ and ‘fully complete‘ but could be improved by designing it differently, provide your feedback and suggest your version of the specification. Or, if the specification is ‘Mostly singular‘, but you would design it differently for any specific reason, explain your feedback and include your version.
6. Is the specification plausible for real-world implementation, given standard physics constraints and common sense (Yes/No)? If not, please elaborate on any concerns or issues that might arise.
7. Ask for our feedback:

   If you have any concerns/questions, please don’t hesitate to reach out to us at []. The quality of your work is our top priority.

Final Note: Your feedback is crucial to evaluating and improving the quality of LLM-generated specifications and ensuring alignment with best practices in BDD. Thank you for your participation!

Observations and Insights: Below are common patterns and suggestions based on our evaluations:

1. Assumptions: LLMs tend to include assumptions. We should decide to what degree their generated content is relevant. Content could be considered relevant and acceptable based on a common-sense interpretation of input or potentially irrelevant, i.e., they contradict the input or are not supported by the input.

   Example:

   Regulatory Requirement: "(5) In the case of an inspection of a consumer prepackaged food that consists of a liquid, the net quantity of the food must be determined on the basis of the assumption that the liquid is at a temperature of 20°C."

   Original Specification:

   "Feature: Liquid Food Product Inspection

   Scenario: Measuring net quantity of prepackaged liquid food

   Given a consumer prepackaged food product containing liquid

   And the liquid temperature is stabilized at 20 degrees Celsius

   When the inspector measures the net quantity

   Then the system should record the measurement value

   And store the measurement with the temperature condition

   Scenario: Attempting measurement at incorrect temperature

   Given a consumer prepackaged food product containing liquid

   And the liquid temperature is not at 20 degrees Celsius

   When the inspector attempts to measure the net quantity

   Then the system should display a warning message

   And prevent the measurement from being recorded

   And prompt for temperature adjustment to 20 degrees Celsius"

   Improved Specification:

   "Feature: Liquid Food Product Inspection

   As a food safety inspector, I need to measure the net quantity of prepackaged liquid food adjusted to 20°C so that I can ensure accurate quantity measurements for consumer protection

   Background: Given the inspection system is calibrated for temperature-adjusted liquid measurements And the system has temperature monitoring and adjustment capabilities

   Scenario: Measuring net quantity of prepackaged liquid food

   Given a consumer prepackaged food product containing liquid

And the liquid temperature is measured at its current state

When the inspector measures the net quantity

Then the system should calculate the adjusted volume as if the liquid is at 20°C

And record the adjusted measurement value

And store the temperature used for adjustment"

Tip: The second scenario introduces a counter-argument with steps not pertinent to the input (e.g., the system displaying a warning message and preventing measurement recording) that could be avoided to improve relevance.

Also, the specification should cover intended functionalities that are plausible. For this regulatory requirement, measuring liquid net quantity at 20°C does not follow common practices. The reason is that liquids, conceivably, would never be at a specific temperature we want, i.e., liquids are not always temperature-controlled. In this case, it is more plausible that the liquid is volume adjusted as if it is at 20 degrees.

2. Details: Sometimes, details usually in the user story or background section and mostly as context, are provided, which are not exact words of the given regulatory requirement, but they are pertinent and valid. These relevant details often help clarify the intent of the feature without compromising relevance. For instance, in a regulatory requirement that addresses the temperature measurement of food products having a background of "Given the inspection system is calibrated for liquid measurements And temperature monitoring capability is available" could be considered a bonus for the model. In these cases, you might want to write down your feedback about the specific details you found acceptable and relevant.

3. Single clear intention:

   Example:

   Regulatory Requirement: "Liquid Whole Egg, Dried Whole Egg and Frozen Whole Egg are the foods that meet the standard set out in section B.22.034 of the FDR and:

   (a) in the case of liquid whole egg and frozen whole egg, contain at least 23.5% egg solids by weight;
   (b) in the case of dried whole egg, contain not more than 5% water; and
   (c) contain not more than 50,000 viable bacteria per gram and 10 coliform organisms per gram."

   Original Specification:

   "Feature: Egg Product Requirements

   Scenario: Liquid Whole Egg requirements

   Given a liquid whole egg product

   When the product is analyzed

Then it must contain at least 23.5% egg solids by weight
And it must not contain more than 50,000 viable bacteria per gram
And it must not contain more than 10 coliform organisms per gram
Scenario: Frozen Whole Egg requirements
Given a frozen whole egg product
When the product is analyzed
Then it must contain at least 23.5% egg solids by weight
And it must not contain more than 50,000 viable bacteria per gram
And it must not contain more than 10 coliform organisms per gram
Scenario: Dried Whole Egg requirements
Given a dried whole egg product
When the product is analyzed
Then it must contain not more than 5% water
And it must not contain more than 50,000 viable bacteria per gram
And it must not contain more than 10 coliform organisms per gram"

Improved Specification:
"Feature: Egg Product Requirements
Background:
Given an egg product ready for analysis
Rule: Egg products must meet the FDR standards for composition and bacteria limits
Scenario: Verify composition for Liquid and Frozen Whole Eggs
Given the product type is liquid whole egg or frozen whole egg
When the product is analyzed
Then it must contain at least 23.5% egg solids by weight
Scenario: Verify composition for Dried Whole Eggs
Given the product type is dried whole egg
When the product is analyzed
Then it must contain not more than 5% water
Scenario: Verify bacteria and coliform limits for all egg products
When the product is analyzed for bacteria content
Then it must not contain more than 50,000 viable bacteria per gram
And it must not contain more than 10 coliform organisms per gram"
Tip: The improved version separates different functions into distinct scenarios, maintaining distinct purposes for each scenario. This improves singularity.
Example:
Regulatory Requirement: "Heating requirements for item 3 a Liquid processed egg, Whole egg with no less than 24% and no more than 38% egg solids is

Minimum temperature of the processed egg at the automatic diversion valve 61 (°C), Minimum temperature of the processed egg at the automatic diversion valve 142 (°F) and Minimum heating time 3.5 (minutes).

Heating requirements for item 4 b Liquid processed egg, Whole egg mix with less than 2% added salt or sweetening agent, or both is Minimum temperature of the processed egg at the automatic diversion valve 60 (°C), Minimum temperature of the processed egg at the automatic diversion valve 140 (°F) and Minimum heating time 6.2 (minutes)."

Original Specification:

"Feature: Heating Requirements for Liquid Processed Eggs

Scenario Outline: Verify heating requirements for liquid processed eggs

Given a liquid processed egg of type "<Egg Type>"

And the egg has "<Egg Solids>" egg solids

When it is heated at the automatic diversion valve

Then the minimum temperature should be "<Temperature (°C)>" degrees Celsius

And the minimum temperature should be "<Temperature (°F)>" degrees Fahrenheit

And the minimum heating time should be "<Heating Time>" minutes

Examples: | Egg Type | Egg Solids | Temperature (°C) | Temperature (°F) | Heating Time | | Whole egg | 24%-38% | 61 | 142 | 3.5 | | Whole egg mix (≤2% salt/sweetener) | N/A | 60 | 140 | 6.2 |"

Improved Specification:

"Feature: Heating Requirements for Liquid Processed Egg

The feature provides behavioral specifications for compliance with regulatory requirements on pasteurization/heating of various liquid processed egg mixtures.

@Requirement-3a

Scenario: Ensure heating requirements for Item 3(a) - Whole egg (24-38% solids)

Given the product is "Liquid processed egg, Whole egg

And the egg solid content is between 24% and 38%

When the system measures temperature at the automatic diversion valve

Then the minimum temperature shall be 61 °C (142 °F)

And the minimum heating time shall be 3.5 minutes

@Requirement-4b

Scenario: Ensure heating requirements for Item 4(b) - Whole egg mix (<2% salt/sweetener)

Given the product is "Liquid processed egg, Whole egg mix"

And less than 2% of salt or sweetening agent (or both) is added

When the system measures temperature at the automatic diversion valve

Then the minimum temperature shall be 60 °C (140 °F) And the minimum heating time shall be 6.2 minutes"

Tip: Splitting scenarios further enhances singularity and testability by isolating distinct functions.

C.2 Content of Recruitment Form

Subject: Invitation to Participate in Research on LLM-Generated Requirements Specifications

Dear uOttawa student,

We are inviting you to participate to a research study aiming to evaluate software artifacts generated by Large Language Models (LLM), specifically focusing on their use in creating Gherkin specifications for regulatory provisions. This research is motivated by real-world applications and seeks to enhance the quality and usability of AI-generated artifacts, compared to those that might be created by knowledgeable software engineers. The study will be conducted solely in English. Therefore, participants must have sufficient proficiency in English to engage effectively with the study materials. Participants will be selected on a first-come, first-served basis.

Details of Participation:
Your participation will involve reviewing and providing feedback on LLM-generated Gherkin specifications through an online Google Sheets format. You are expected to commit 12 tasks. If you complete all tasks, you will receive $8 per task and an additional completion bonus of $54, which in total is $150. If you withdraw at any part in the project, you will receive $8 per task you have completed.

Procedure:
1) Review LLM-Generated Artifacts: Access a set of Gherkin specifications generated by LLMs via a provided Google Sheets link.

2) Provide Feedback: Evaluate each specification for accuracy and completeness, and suggest necessary revisions.

3) Submit Feedback: Enter your responses directly into Google Sheets, ensuring an efficient process.

Benefits of Participating:

1) Gain firsthand experience with reviewing requirements and test cases in Agile development,

2) Contribute to advancements in the field of AI-enabled software engineering,

3) Receive compensation for your valuable insights and time.

Confidentiality:
Your participation will be completely anonymous, ensuring that no personal data or responses can be traced back to you in any published material.

Voluntary Participation:
Participation in this study is entirely voluntary. You may withdraw at any time without any consequences. If you are interested in participating, please indicate your interest by replying to Shabnam Hassani []. In your email, please explain how you are meeting the qualifications in terms of familiarity with Agile development, BDD and software engineering. For any questions or further information, please feel free to contact us at [].

Thank you for considering this opportunity to advance our understanding of AI's role in software engineering.

Best regards,
Mehrdad Sabetzadeh, Daniel Amyot, Shabnam Hassani University of Ottawa

C.3 Content of the Consent Form

Project Name: Assessing the Quality of Automatically Generated Systems and Software Requirements by Large Language Models

Principal Investigator:
Mehrdad Sabetzadeh, Professor

Co-Investigator:
Daniel Amyot, Professor

Research Assistant:
Shabnam Hassani, PhD Candidate

Introduction:
Before agreeing to participate in this research project, please take the time to read and carefully consider the following information. This document explains the purpose of the research project, its procedures, risks, and benefits.
You are invited to participate in a research study aiming to evaluate software artifacts generated by Large Language Models (LLM), specifically focusing on Gherkin specifications for food-safety monitoring systems. This project is conducted by the University of Ottawa's research team led by Profs Mehrdad Sabetzadeh and Daniel Amyot. The study is funded by NSERC through grants RGPIN-2020-03991 and RGPAS-2020-00076.

Study Description:
You will review and provide feedback on LLM-generated Gherkin specifications using an online Google Sheets format. The expected duration of your participation is approximately 6 hours, divided as needed to suit your schedule, as long as you can finish your tasks within 10 days.

Consent Process:
To participate in this study, please read this document carefully and provide your consent by submitting a confirmation through the provided online form. There is no need for verbal or audio consent as all communications and submissions are handled electronically.

Confidentiality:
Your participation in this study will be completely anonymous, and no personal data or responses will be traceable back to you in any published material.

Voluntary Participation:
Your participation is entirely voluntary, and you may withdraw at any time. If you choose to withdraw, any data you have provided will be excluded from the study.

Compensation:
You will be compensated based on the completion of specific tasks. Specifically, you will receive $8 for each Gherkin scenario evaluation task you begin, with a total of 12 tasks available. If you complete all tasks, you will receive a total of $96. Also, a completion bonus of $54 will be provided to participants who complete all 12 tasks, bringing the total potential compensation to $150.
If you choose to withdraw from the study before completing all assigned tasks, you will still receive compensation for the tasks you have started. Specifically, you will receive $8 for each task you begin, regardless of whether you complete the entire set of tasks. However, you will not be eligible for the $54 completion bonus unless all tasks are completed.
Compensation will be provided upon the completion of the study or upon withdrawal, based on the number of tasks started. The method of compensation will be agreed upon in this consent document and may include options such as direct deposit or university account credit.

Risks and Benefits:
There are no known risks associated with participating in this study. Participants may benefit from the experience by gaining insights into requirements specification, agile development, and legal compliance.

Further Information and Contact:
If you have questions at any time about the study or the procedures, you may contact the research supervisors at the email addresses above.

Ethics Approval:
REB has approved the ethical components of this study. Participants should print a

copy of the consent form to keep for their personal records.
If you have any concerns about your rights as a participant or the ethical conduct of this study, please contact:
The Protocol Officer for Ethics in Research, University of Ottawa

Confirmation:
By checking the box "I agree to participate" below, you indicate that you have understood the information and agree to participate in this research study.
By checking the box “I agree to participate” below, you indicate that you have understood the information and agree to participate in this research study.

☐ I agree to participate.

☐ I do not wish to participate.

Participant’s name and signature: _____ **Date:** _____

Researcher’s name and signature: _____ **Date:** _____

C.4 Supplementary Material for Section 5.5

Example 3 – Legal Provision

Before slaughtering a food animal that is an equine or a bird, the holder of a licence to slaughter must obtain, from the person who owned or had the possession, care or control of the food animal before its arrival at the establishment where it is intended to be slaughtered, documents that include the following information:

(a) the name and contact information of the person who owned it and any person who had the possession, care or control of it immediately before its arrival at the establishment;

(b) the last location where it was raised or kept before its arrival at the establishment, including the address of the location, its code or the number that identifies it;

(c) the food animal’s identification number or any other information that identifies it;

(d) the name of any on-farm food safety program under which the food animal was raised or kept;

(e) in the case of a bird,

(i) the time at which the first bird of the lot of birds was placed into a crate,

(ii) the time at which the bird was last provided with access to a source of hydration before being loaded, and

(iii) the time at which the bird was last provided with access to feed before being loaded;

(f) a description of any physical or chemical hazard to which the food animal may have been exposed that could cause any meat product that is derived from the food animal to be contaminated;

(g) in respect of the last 120 days of the life of a bird that has been used for breeding or egg production or in respect of the entire life of any other bird,

(i) the mortality rate for the flock from which the bird originates,

(ii) the name of any disease or syndrome that was diagnosed in the flock from which the bird originates and the date on which the flock recovered from the disease or syndrome,

(iii) the names of any drugs administered to treat a disease or syndrome or administered as an extra-label use and of any vaccines that were administered to the bird, as well as

(A) their route of administration,

(B) the first and last date of their administration,

(C) the dosage that was administered, and

(D) the withdrawal period or, in the case of an extra-label drug administration, a copy of the prescription that was issued by a veterinarian and an attestation by a competent person or body with respect to the withdrawal period for that administration,

(iv) the name of any drug that has been administered in the last 14 days that requires a withdrawal period;

(h) in respect of the last 180 days of the life of an equine,

(i) the name of any disease or syndrome that was diagnosed or a description of any deviation from normal behaviour, physiology or appearance,

(ii) the names of any drugs and vaccines that were administered to the equine, as well as

(A) their Drug Identification Numbers, if any,

(B) their route of administration,

(C) the last date of their administration,

(D) the dosage that was administered, and

```
Feature: Slaughterhouse Pre-Slaughter Documentation
As a slaughterhouse operator
I want to ensure that I have the required pre-slaughter documentation
    So that I can comply with food safety regulations
Scenario: Equine Pre-Slaughter Documentation
Given a food animal is an equine
When I receive the equine at the slaughterhouse
Then I must obtain documents with the following information:


| Information                | Description                                    |
| Owner name and contact     | Name and contact information of the owner      |
| Previous owner/possessor   | Name and contact information of previous       |
| Last location              | Address, code, or identifier of the last location  |
| Equine ID                  | Equine identification number or other identifying information    |
| On-farm food safety program | Name of the on-farm food safety program       |
| Physical/chemical hazards  | Description of any physical or chemical hazards    |
| Health history             | Health history for the last 180 days, including:   |
|                            |  - Diseases or syndromes diagnosed                 |
|                            |  - Deviations from normal behavior, physiology,or appearance |
|                            |  - Drugs and vaccines administered, including:     |
|                            |  - Drug Identification Numbers (if applicable)     |
|                            |  - Route of administration                         |
|                            |  - Last date of administration                     |
|                            |  - Dosage administered                             |
|                            |  - Withdrawal period                               |
| Equine use                 | Use of the equine                                  |


Scenario: Bird Pre-Slaughter Documentation
Given a food animal is a bird
When I receive the bird at the slaughterhouse
Then I must obtain documents with the following information:


| Information                | Description                                                  |
| Owner name and contact     | Name and contact information of the owner                    |
| Previous owner/possessor   | Name and contact information of previous owner/possessor     |
| Last location              | Address, code, or identifier of the last location            |
| Bird ID                    | Bird identification number or other identifying information  |
| On-farm food safety program | Name of the on-farm food safety program                      |
| Physical/chemical hazards  | Description of any physical or chemical hazards              |
| Time of crating            | Time the first bird of the lot was placed into a crate       |
| Time of last hydration     | Time the bird was last provided with access to hydration     |
| Time of last feeding       | Time the bird was last provided with access to feed          |
| Health history             | Health history for the last 120 days (breeding/egg production) or entire life (other birds), including: |
|                            |  - Mortality rate for the flock                                |
|                            |  - Diseases or syndromes diagnosed                             |
|                            |  - Drugs and vaccines administered, including:                 |
|                            |  - Route of administration                                     |
|                            |  - First and last date of administration                       |
|                            |  - Dosage administered                                         |
|                            |  - Withdrawal period                                           |
| Recent medication          | Name of any drug administered in the last 14 days requiring a withdrawal period |

```

Figure C.1: LLM-derived Gherkin specification for Example 3.

(E) the withdrawal period or, in the case of an extra-label drug administration, a copy of the prescription that was issued by a veterinarian and an attestation by a competent person or body with respect to the withdrawal period for that administration,

(iii) the use of the equine.

```gherkin
# EQUINES
Feature: Pre-slaughter documentation – Equine
Background:
  Given an animal of type "equine" arrives for slaughter
  And the operator starts the pre-slaughter documentation workflow

# (a) Owner / custodian
Scenario: Record equine owner and custodian contact details
When the operator enters ownership information
Then the documentation must include:
  | owner.name |
  | owner.contact_information  |
  | custodian.name |
  | custodian.contact_information |
# (b) Last premises
Scenario: Record last premises of the equine
When the operator enters origin information
Then the documentation must include one of:
  | last_premises.address |
  | last_premises.location_code |
  | last_premises.identifier |
# (c) Animal identifier
Scenario: Record equine identification number
When the operator enters identity information
Then the documentation must include:
  | equine.id |
  | equine.alt_id? |
# (d) On-farm food-safety programme
Scenario: Record on-farm food-safety programme (equine)
When the operator enters farm-programme information
Then the documentation must include:
  | farm_programme.name |
# (f) Possible hazards
Scenario: Record potential physical or chemical hazards (equine)
When the operator reviews exposure history
Then the documentation must include:
  | hazards.description |
# (h) Health history, medications and use
Scenario: Record equine health history and intended use (last 180 days)
When the operator enters health-history information
Then the documentation must include:
  | disease_or_syndrome.name |
  | disease_or_syndrome.diagnosis_date |
  | disease_or_syndrome.recovery_date? |
  | deviation.description? |
  | drug_or_vaccine.identification_number? |
  | drug_or_vaccine.route |
  | drug_or_vaccine.last_admin_date |
  | drug_or_vaccine.dosage |
  | drug_or_vaccine.withdrawal_period |
  | drug_or_vaccine.extra_label_prescription? |
  | drug_or_vaccine.extra_label_attest? |
  | equine_use |
```

Figure C.2: Revised Gherkin specification for Example 3 (first part).

```
# BIRDS
Feature: Pre-slaughter documentation – Bird
Background:
Given an animal of type "bird" arrives for slaughter
And the operator starts the pre-slaughter documentation workflow

# (a) Owner / custodian
Scenario: Record bird owner and custodian contact details
When the operator enters ownership information
Then the documentation must include:
| owner.name            | owner.contact_information  | custodian.name          | custodian.contact_information |
# (b) Last premises
Scenario: Record last premises of the bird
When the operator enters origin information
Then the documentation must include one of:
| last_premises.address         | last_premises.location_code  | last_premises.identifier   |
# (c) Animal identifier
Scenario: Record bird identification number
When the operator enters identity information
Then the documentation must include:
| bird.id   | bird.alt_id?  |
# (d) On-farm food-safety programme
Scenario: Record on-farm food-safety programme (bird)}
When the operator enters farm-programme information
Then the documentation must include:
| farm_programme.name |
# (e) Crating, hydration and feeding times
Scenario: Record bird handling times before loading
When the operator enters bird-handling information
Then the documentation must include:
| time_first_bird_crated | time_last_hydration  | time_last_feeding  |
# (f) Possible hazards
Scenario: Record potential physical or chemical hazards (bird)
When the operator reviews exposure history
Then the documentation must include:
| hazards.description |
# (g) (i) and (ii) Flock health
Scenario: Record flock health and medication history (last 120 days or entire life)
When the operator enters health-status information
Then the documentation must include:
| flock.mortality_rate  | disease_or_syndrome.name       | disease_or_syndrome.recovery_date? |
# (g)(iii) Drugs & vaccines
Scenario: Record bird medications and vaccines (last 120 days or entire life)
When the operator enters medication history
Then the documentation must include for each drug_or_vaccine:
| drug_or_vaccine.route | drug_or_vaccine.first_admin_date   | drug_or_vaccine.last_admin_date | drug_or_vaccine.dosage     |
drug_or_vaccine.withdrawal_period | drug_or_vaccine.extra_label_prescription?     | drug_or_vaccine.extra_label_attest?    |
# (g)(iv) Drugs within last 14 days
Scenario: Record drugs given in last 14 days that require withdrawal
When the operator enters recent medication history
Then the documentation must include for each recent_drug:
| recent_drug.name    | recent_drug.withdrawal_period  |
```

Figure C.3: Revised Gherkin specification for Example 3 (second part).

References

[1] Muhammad Abbas, Alessio Ferrari, Anas Shatnawi, Eduard Enoiu, Mehrdad Saadatmand, and Daniel Sundmark. On the relationship between similar requirements and similar software: A case study in the railway domain. *Requirements Engineering*, 28(1):23–47, 2023. doi:10.1007/s00766-021-00370-4.

[2] Sallam Abualhaija, Chetan Arora, Mehrdad Sabetzadeh, Lionel C Briand, and Michael Traynor. Automated demarcation of requirements in textual specifications: a machine learning-based approach. *Empirical Software Engineering*, 25:5454–5497, 2020. doi:10.1007/s10664-020-09864-1.

[3] Sallam Abualhaija, Chetan Arora, Amin Sleimi, and Lionel C Briand. Automated question answering for improved understanding of compliance requirements: A multi-document study. In *Proceedings of 30th IEEE International Requirements Engineering Conference (RE)*, pages 39–50. IEEE, 2022. doi:10.1109/RE54965.2022.00011.

[4] 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 *Proceedings of 32nd IEEE International Requirements Engineering Conference (RE)*, pages 5–17. IEEE, 2024. doi:10.1109/RE59067.2024.00012.

[5] Sallam Abualhaija, Marcello Ceci, Nicolas Sannier, Domenico Bianculli, Salomé Lannier, Martina Siclari, Olivier Voordeckers, and Stanislaw Tosza. LLM-assisted extraction of regulatory requirements: A case study on the GDPR. In *33rd IEEE International Requirements Engineering Conference (RE)*. IEEE CS, 2025. https://hdl.handle.net/10993/65265.

[6] Toufique Ahmed, Premkumar Devanbu, Christoph Treude, and Michael Pradel. Can LLMs replace manual annotation of software engineering artifacts? In *2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)*, pages 526–538. IEEE, 2025. doi:10.1109/MSR66628.2025.00086.

[7] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, page 2623–2631. Association for Computing Machinery, 2019. doi:10.1145/3292500.3330701.

[8] Saranya Alagarsamy, Chakkrit Tantithamthavorn, Wannita Takerngsaksiri, Chetan Arora, and Aldeida Aleti. Enhancing large language models for text-to-testcase generation. *Journal of Systems and Software*, 230:112531, 2025. doi:10.1016/j.jss.2025.112531.

[9] Waad Alhoshan, Liping Zhao, Alessio Ferrari, and Keletso J Letsholo. A zero-shot learning approach to classifying requirements: A preliminary study. In *Proceedings of 28th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ)*, pages 52–59. Springer, 2022. doi:10.1007/978-3-030-98464-9_5.

[10] Orlando Amaral, Sallam Abualhaija, and Lionel Briand. ML-based compliance verification of data processing agreements against GDPR. In *Proceedings of 31st IEEE International Requirements Engineering Conference (RE)*, pages 53–64. IEEE, 2023. doi:10.1109/RE57278.2023.00015.

[11] Orlando Amaral, Sallam Abualhaija, Mehrdad Sabetzadeh, and Lionel Briand. A Model-based conceptualization of requirements for compliance checking of data processing against GDPR. In *Proceedings of 29th International Requirements Engineering Conference Workshops (REW)*, pages 16–20. IEEE, 2021. doi:10.1109/REW53955.2021.00009.

[12] Orlando Amaral, Sallam Abualhaija, Damiano Torre, Mehrdad Sabetzadeh, and Lionel C. Briand. AI-enabled automation for completeness checking of privacy policies. *IEEE Transactions on Software Engineering*, 48(11):4647–4674, 2021. doi:10.1109/TSE.2021.3124332.

[13] Orlando Amaral, 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*, pages 4282 – 4303, 2023. doi:10.1109/TSE.2023.3288901.

[14] Amazon Web Services. Meta's Llama 3.3 70b model now available in Amazon Bedrock, 2024. https://aws.amazon.com/about-aws/whats-new/2024/12/metas-llama-3-3-70b-model-amazon-bedrock/.

[15] Florian Angermeir, Jannik Fischbach, Fabiola Moyón, and Daniel Mendez. Towards automated continuous security compliance. In *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '24, page 440–446. ACM, 2024. doi:10.1145/3674805.3690748.

[16] Preethu Rose Anish, Aparna Verma, Sivanthy Venkatesan, Logamurugan V, and Smita Ghaisas. Governance-focused classification of security and privacy requirements from obligations in software engineering contracts. In *Proceedings of 30th International Working Conference on Requirement Engineering: Foundation for Software Quality (REFSQ)*, page 92–108. Springer-Verlag, 2024. doi:10.1007/978-3-031-57327-9_6.

[17] Anthropic. Claude, 2023. https://www.anthropic.com/claude.

[18] Anthropic. Prompting 101 | code w/ claude, 2025. URL: https://bit.ly/4p3TPu5.

[19] Annie I Antón and Julia B Earp. A requirements taxonomy for reducing web site privacy vulnerabilities. *Requirements Engineering*, 9(3):169–185, 2004. doi:10.1007/s00766-003-0183-z.

[20] Archer. Archer integrated risk management platform. https://www.archerirm.com/.

[21] Andrea Arcuri and Lionel Briand. A practical guide for using statistical tests to assess randomized algorithms in software engineering. In *Proceedings of the 33rd International Conference on Software Engineering*, ICSE '11, page 1–10. ACM, 2011. doi:10.1145/1985793.1985795.

[22] Chetan Arora, John Grundy, and Mohamed Abdelrazek. *Advancing Requirements Engineering Through Generative AI: Assessing the Role of LLMs*, chapter 6 of Generative AI for Effective Software Development, pages 129–148. Springer, 2024. doi:10.1007/978-3-031-55642-5_6.

[23] Chetan Arora, Mehrdad Sabetzadeh, Lionel Briand, and Frank Zimmer. Automated extraction and clustering of requirements glossary terms. *IEEE Transactions on Software Engineering*, 43(10):918–945, 2016. doi:10.1109/TSE.2016.2635134.

[24] Atlassian. User Story, 2024. https://www.atlassian.com/agile/project-management/user-stories [last accessed: Dec. 2024].

[25] Sebastian Baltes et al. Guidelines for empirical studies in software engineering involving large language models. *arXiv:2508.15503*, 2025. doi:10.48550/arXiv.2508.15503.

[26] Victor R Basili, Gianluigi Caldiera, and H Dieter Rombach. Goal, question, metric paradigm. *Encyclopedia of software engineering*, 1:646–661, 1994. URL: https://www.cs.umd.edu/~basili/publications/technical/T89.pdf.

[27] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1), 1995.

[28] Armin Berger, Lars Hillebrand, David Leonhard, Tobias Deußer, Thiago Bell Felix De Oliveira, Tim Dilmaghani, Mohamed Khaled, Bernd Kliem, Rudiger Loitz, Christian Bauckhage, et al. Towards automated regulatory compliance verification in financial auditing with large language models. In *2023 IEEE International Conference on Big Data (BigData)*, pages 4626–4635. IEEE, 2023. doi:10.1109/BigData59044.2023.10386518.

[29] Jaspreet Bhatia and Travis D Breaux. Semantic incompleteness in privacy policy goals. In *Proceedings of 26th IEEE International Requirements Engineering Conference (RE)*, pages 159–169. IEEE, 2018. doi:10.1109/RE.2018.00025.

[30] Leonard Peter Binamungu, Suzanne M Embury, and Nikolaos Konstantinou. Characterising the quality of behaviour driven development specifications. In *Agile Processes in Software Engineering and Extreme Programming*, pages 87–102. Springer International Publishing, 2020. doi:10.1007/978-3-030-49392-9_6.

[31] Leonard Peter Binamungu and Salome Maro. Behaviour driven development: A systematic mapping study. *Journal of Systems and Software*, 203:111749, 2023. doi:10.1016/j.jss.2023.111749.

[32] Léon Bottou. Online learning and stochastic approximations. *On-line Learning in Neural Networks*, 17(9), 1998. URL: https://leon.bottou.org/publications/pdf/online-1998.pdf.

[33] Yamine Bouzembrak, Marcel Klüche, Anand Gavai, and Hans J.P. Marvin. Internet of Things in food safety: Literature review and a bibliometric analysis. *Trends in Food Science & Technology*, 94:54–64, 2019. doi:10.1016/j.tifs.2019.11.002.

[34] Travis Breaux and Annie Antón. Analyzing regulatory rules for privacy and security requirements. *IEEE Transactions on Software Engineering*, 34(1):5–20, 2008. doi:10.1109/TSE.2007.70746.

[35] Travis D. Breaux and David G. Gordon. Preserving traceability and encoding meaning in legal requirements extraction. In *Proceedings of 6th International Workshop on Requirements Engineering and Law (RELAW)*, pages 57–60. IEEE, 2013. doi:10.1109/RELAW.2013.6671347.

[36] Travis D. Breaux, Matthew W. Vail, and Annie I. Antón. Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. In *14th IEEE International Requirements Engineering Conference (RE'06)*, pages 49–58. IEEE CS, 2006. doi:10.1109/RE.2006.68.

[37] Travis Durand Breaux. *Legal requirements acquisition for the specification of legally compliant information systems*. PhD thesis, North Carolina State University, USA, 2009. URL: http://www.lib.ncsu.edu/resolver/1840.16/3376.

[38] Steve Brewer, Simon Pearson, Roger Maull, Phil Godsiff, Jeremy G Frey, Andrea Zisman, Gerard Parr, Andrew McMillan, Sarah Cameron, Hannah Blackmore, et al. A trust framework for digital food systems. *Nature Food*, 2(8):543–545, 2021. doi:10.1038/s43016-021-00346-1.

[39] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 2020. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[40] David Campbell and Philip Thomas. *Fundamental legal conceptions as applied in judicial reasoning by Welsey Newcomb Hohfeld*. Routledge, 2016.

[41] J. Anthony Capon. *Elementary Statistics for the Social Sciences: Study Guide*. Wadsworth, 1991.

[42] Kathy Charmaz. *Constructing grounded theory*, volume 10. sage, 2014. doi:10.1177/1363459306067319.

[43] Ranit Chatterjee, Abdul Ahmed, Preethu Rose Anish, Brijendra Suman, Prashant Lawhatre, and Smita Ghaisas. A pipeline for automating labeling to prediction in classification of NFRs. In *2021 IEEE 29th International Requirements Engineering Conference (RE)*, pages 323–323. IEEE CS, 2021. doi:10.1109/RE51729.2021.00036.

[44] Boyuan Chen, Mingzhi Wen, Yong Shi, Dayi Lin, Gopi Krishnan Rajbahadur, and Zhen Ming Jiang. Towards training reproducible deep learning models. In *Proceedings of 44th International Conference on Software Engineering (ICSE)*, pages 2202–2214. ACM, 2022. doi:10.1145/3510003.3510163.

[45] Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. Walking down the memory maze: Beyond context limit through interactive reading, 2023. arXiv:2310.05029.

[46] Antonio Cordella and Francesco Gualdi. Regulating generative AI: The limits of technology-neutral regulatory frameworks. insights from Italy's intervention on ChatGPT. *Government Information Quarterly*, 41(4):101982, 2024. doi:10.1016/j.giq.2024.101982.

[47] Harsh Dadhaneeya, Prabhat K Nema, and Vinkel Kumar Arora. Internet of Things in food processing and its potential in Industry 4.0 era: A review. *Trends in Food Science & Technology*, 139:104109, 2023. doi:10.1016/j.tifs.2023.07.006.

[48] Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel Ho. Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, 16(1):64–93, 2024. doi:10.1093/jla/laae003.

[49] Souvick Das, Novarun Deb, Agostino Cortesi, and Nabendu Chaki. Zero-shot learning for named entity recognition in software specification documents. In *Proceedings of 31st IEEE International Requirements Engineering Conference (RE)*, pages 100–110. IEEE, 2023. doi:10.1109/RE57278.2023.00019.

[50] Joost FC de Winter and Dimitra Dodou. Five-point Likert items: t test versus Mann-Whitney-Wilcoxon (Addendum added October 2012). *Practical Assessment, Research, and Evaluation*, 15(1):11, 2010. doi:10.7275/bj1p-ts64.

[51] Gouri Deshpande, Behnaz Sheikhi, Saipreetham Chakka, Dylan Lachou Zotegouon, Mohammad Navid Masahati, and Guenther Ruhe. Is BERT the new silver bullet? -

An empirical investigation of requirements dependency classification. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pages 136–145. IEEE CS, 2021. doi:10.1109/REW53955.2021.00025.

[52] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of 17th American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, 2019. doi:10.18653/v1/n19-1423.

[53] DocsBot AI. Llama 3.3 70b Instruct vs Llama 3.1 405B Instruct - detailed performance & feature comparison, 2024. https://docsbot.ai/models/compare/llama-3-3-70b-instruct/llama-3-1-405b-instruct.

[54] George T Doran. There's a S.M.A.R.T. way to write management's goals and objectives. *Management Review*, 70(11):35, 1981. URL: https://bit.ly/S-M-A-R-T-1981.

[55] Ernani César dos Santos and Patrícia Vilain. Automated acceptance tests as software requirements: An experiment to compare the applicability of Fit Tables and Gherkin Language. In *Agile Processes in Software Engineering and Extreme Programming*, pages 104–119. Springer, 2018. doi:10.1007/978-3-319-91602-6_7.

[56] Parisa Elahidoost, Daniel Mendez, Michael Unterkalmsteiner, Jannik Fischbach, Christian Feiler, and Jonathan Streit. Practices, challenges, and opportunities when inferring requirements from regulations in the FinTech sector - An industrial study. In *2024 IEEE 32nd International Requirements Engineering Conference Workshops (REW)*, pages 137–145. IEEE CS, 2024. doi:10.1109/REW61692.2024.00024.

[57] European Union. Regulation (eu) 2016/679 of the european parliament and of the council 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, and repealing directive 95/46/ec (general data protection regulation). *Official Journal of the European Union*, L119:1–88, 2016. OJ L 119, 4.5.2016. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[58] Morgan C. Evans, Jaspreet Bhatia, Sudarshan Wadkar, and Travis D. Breaux. An evaluation of constituency-based hyponymy extraction from privacy policies. In *Proceedings of 25th IEEE International Requirements Engineering Conference (RE)*, pages 312–321. IEEE, 2017. doi:10.1109/RE.2017.87.

[59] Saad Ezzini, Sallam Abualhaija, and Mehrdad Sabetzadeh. WikiDoMiner: Wikipedia domain-specific miner. In *Proceedings of 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, pages 1706–1710, 2022. doi:10.5281/zenodo.6671357.

[60] Andreas Falkner, Cristina Palomares, Xavier Franch, Gottfried Schenner, Pablo Aznar, and Alexander Schoerghuber. Identifying requirements in requests for proposal:

A research preview. In *Proceedings of 25th International Working Conference on Requirement Engineering: Foundation for Software Quality (REFSQ)*, pages 176–182. Springer, 2019. doi:10.1007/978-3-030-15538-4_13.

[61] Ming Fan, Le Yu, Sen Chen, Hao Zhou, Xiapu Luo, Shuyue Li, Yang Liu, Jun Liu, and Ting Liu. An empirical evaluation of GDPR compliance violations in Android mHealth apps. In *Proceedings of 31st IEEE international symposium on software reliability engineering (ISSRE)*. IEEE, 2020. doi:10.1109/ISSRE5003.2020.00032.

[62] Alessandro Fantechi, Stefania Gnesi, Lucia Passaro, and Laura Semini. Inconsistency detection in natural language requirements using chatgpt: a preliminary evaluation. In *2023 IEEE 31st International Requirements Engineering Conference (RE)*, pages 335–340. IEEE, 2023. doi:10.1109/RE57278.2023.00045.

[63] Alessandro Fantechi, Stefania Gnesi, and Laura Semini. Exploring LLMs' ability to detect variability in requirements. In *Proceedings of 30th International Working Conference on Requirement Engineering: Foundation for Software Quality (REFSQ)*, pages 178–188. Springer, 2024. doi:10.1007/978-3-031-57327-9_11.

[64] Mohamad Fazelnia, Viktoria Koscinski, Spencer Herzog, and Mehdi Mirakhorli. Lessons from the use of natural language inference (NLI) in requirements engineering tasks. In *Proceedings of 32nd IEEE International Requirements Engineering Conference (RE)*, pages 103–115. IEEE CS, 2024. doi:10.1109/RE59067.2024.00020.

[65] Alessio Ferrari and Paola Spoletini. Formal requirements engineering and large language models: A two-way roadmap. *Information and Software Technology*, 181:107697, 2025. doi:10.1016/j.infsof.2025.107697.

[66] Margarida Ferreira, Luis Viegas, Joao Pascoal Faria, and Bruno Lima. Acceptance Test Generation with Large Language Models: An Industrial Case Study . In *2025 IEEE/ACM International Conference on Automation of Software Test (AST)*, pages 1–11. IEEE CS, 2025. URL: https://doi.ieeecomputersociety.org/10.1109/AST66626.2025.00007, doi:10.1109/AST66626.2025.00007.

[67] Jannik Fischbach, Julian Frattini, Arjen Spaans, Maximilian Kummeth, Andreas Vogelsang, Daniel Mendez, and Michael Unterkalmsteiner. Automatic detection of causality in requirement artifacts: The CiRa approach. In *Proceedings of 27th International Working Conference on Requirements Engineering: Foundation for software quality (REFSQ)*. Springer, 2021. doi:10.48550/arXiv.2101.10766.

[68] Jannik Fischbach, Julian Frattini, Andreas Vogelsang, Daniel Mendez, Michael Unterkalmsteiner, Andreas Wehrle, Pablo Restrepo Henao, Parisa Yousefi, Tedi Juricic, Jeannette Radduenz, et al. Automatic creation of acceptance tests by extracting conditionals from requirements: NLP approach and case study. *Journal of Systems and Software*, 197:111549, 2023. doi:10.1016/j.jss.2022.111549.

[69] Julian Frattini, Davide Fucci, Richard Torkar, Lloyd Montgomery, Michael Unterkalmsteiner, Jannik Fischbach, and Daniel Mendez. Applying bayesian data analysis for causal inference about requirements quality: a controlled experiment. *EMSE*, 30(1):29, 2025. doi:10.1007/s10664-024-10582-1.

[70] Stefan Fuchs, Michael Witbrock, Johannes Dimyadi, and Robert Amor. Using large language models for the interpretation of building regulations. *Journal of Engineering, Project, and Production Management*, 14(4):0035, 2024. doi:10.32738/JEPPM-2024-0035.

[71] Eduardo C Garrido-Merchan, Roberto Gozalo-Brizuela, and Santiago Gonzalez-Carvajal. Comparing BERT against traditional machine learning models in text classification. *Journal of Computational and Cognitive Engineering*, 2(4), 2023. doi:10.47852/bonviewJCCE3202838.

[72] GFSI. Global food security index 2022, 2022. https://impact.economist.com/sustainability/project/food-security-index/ [last accessed: Dec. 2024].

[73] Sepideh Ghanavati, Daniel Amyot, and Liam Peyton. Towards a framework for tracking legal compliance in healthcare. In *Proceedings of 19th International Conference on Advanced Information Systems Engineering (CAiSE)*, pages 218–232. Springer, 2007. doi:10.1007/978-3-540-72988-4_16.

[74] Alasdair Gilchrist. *Industry 4.0: The Industrial Internet of Things*. Apress, 2016.

[75] Barney G Glaser. *Basics of grounded theory analysis: Emergence vs forcing*. Sociology press, 1992.

[76] Barney G Glaser and Anselm L Strauss. *Discovery of grounded theory: Strategies for qualitative research*. Aldine de Gruyter, 1967.

[77] Tuba Gokhan, Kexin Wang, Iryna Gurevych, and Ted Briscoe. RIRAG: Regulatory information retrieval and answer generation, 2024. doi:10.48550/arXiv.2409.05677.

[78] Google.com. Gemini, 2024. https://bit.ly/4cVCcHk.

[79] Binnur Görner and Fatma Başak Aydemir. Generating requirements elicitation interview scripts with Large Language Models. In *Proceedings of the 31st IEEE International Requirements Engineering Conference Workshops (REW)*, pages 44–51. IEEE CS, 2023. doi:10.1109/REW57809.2023.00015.

[80] Government of Canada. Food and drugs act, 1985. https://laws-lois.justice.gc.ca/eng/acts/f-27/ [last accessed: Dec. 2024].

[81] Government of Canada. Food-specific requirements and guidance, 1997. https://bit.ly/CanadaFSRG [Last accessed: Jan, 2024].

[82] Government of Canada. Safe food for Canadians act, 2012. https://laws-lois.justice.gc.ca/eng/acts/S-1.1/ [Last accessed: Dec. 2024].

[83] Government of Canada. Safe food for Canadians regulations, 2018. https://laws-lois.justice.gc.ca/eng/regulations/SOR-2018-108/index.html [Last accessed: Dec 2024].

[84] Government of Canada. Understanding the safe food for Canadians regulations – a handbook for food businesses, 2018. https://bit.ly/SafeFood2018 [Last accessed: Dec. 2024].

[85] Government of U.S. Food and Drug Administration, 1906. https://www.fda.gov/ [last accessed: Dec. 2024].

[86] Government of U.S. Codex alimentarius, 1993. https://www.fao.org/fao-who-codexalimentarius/codex-texts/guidelines/en/ [last accessed: Dec. 2024].

[87] Government of U.S. Codex Alimentarius Codes of Practice, 2024. https://www.fao.org/fao-who-codexalimentarius/codex-texts/codes-of-practice/en/ [last accessed: Dec. 2024].

[88] Government of U.S. Codex Alimentarius Guidelines, 2024. https://www.fao.org/fao-who-codexalimentarius/codex-texts/guidelines/en/ [last accessed: Dec. 2024].

[89] Government of U.S. Codex Alimentarius Standards, 2024. https://www.fao.org/fao-who-codexalimentarius/codex-texts/list-standards/en/ [last accessed: Dec. 2024].

[90] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings of 2005 IEEE International Joint Conference on Neural Networks (IJCNN)*, volume 4, pages 2047–2–052. IEEE, 2005. doi:10.1109/IJCNN.2005.1556215.

[91] Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, et al. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in neural information processing systems*, 36:44123–44279, 2023.

[92] Mohammad Kasra Habib, Stefan Wagner, and Daniel Graziotin. Detecting requirements smells with deep learning: Experiences, challenges and future work. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pages 153–156. IEEE CS, 2021. doi:10.1109/REW53955.2021.00027.

[93] 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 18th International Conference on Artificial Intelligence and Law (ICAIL)*, pages 40–49, 2021. doi:10.1145/3462757.3466081.

[94] Walid Hariri. Unlocking the potential of ChatGPT: A comprehensive exploration of its applications. *Technology*, 15(2), 2023. doi:10.48550/arXiv.2304.02017.

[95] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G Shin, and Karl Aberer. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 531–548. USENIX Association, 2018. doi:10.5555/3277203.3277243.

[96] Shabnam Hassani. Enhancing legal compliance and regulation analysis with large language models. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 507–511. IEEE, 2024. doi:10.1109/RE59067.2024.00065.

[97] Shabnam Hassani. Full Replication Package: Classification of requirements-related provisions in food-safety regulations: An LLM-based approach, 2024. https://github.com/LLM-basedLegalCompliance/Food-Safety [last accessed: Dec. 2024].

[98] Shabnam Hassani. Online repository: Dataset, 2024. https://bit.ly/4cNQutE.

[99] Shabnam Hassani. Online Repository: Implementation Artifacts, 2024. https://bit.ly/3U7e5y7.

[100] Shabnam Hassani. Online Repository: Rethinking Legal Compliance Automation: Opportunities with Large Language Models, 2024. https://bit.ly/49pbDr0 [Last accessed: Dec. 2024].

[101] Shabnam Hassani. Replication Package: Datasets, 2024. https://github.com/LLM-basedLegalCompliance/Food-Safety/tree/main/Data [Last accessed: Dec. 2024].

[102] Shabnam Hassani. Replication Package: Evaluation Results, 2024. https://github.com/LLM-basedLegalCompliance/Food-Safety/tree/main/Evaluation%20Results [Last accessed: Dec. 2024].

[103] Shabnam Hassani. Replication Package: Implementation Artifacts, 2024. https://github.com/LLM-basedLegalCompliance/Food-Safety/tree/main/Code [Last accessed: Dec. 2024].

[104] Shabnam Hassani. Complete Replication Package on GitHub: From Legal Texts to Gherkin: A Quasi-Experiment on the Quality of LLM-Derived Behavioural Specifications from Regulations, 2025. https://github.com/shabnamhasani/GherkinSpec [Last accessed: June. 2025].

[105] Shabnam Hassani. Replication Package: Datasets, 2025. https://github.com/shabnamhasani/GherkinSpec/tree/main/Data [Last accessed: May. 2025].

[106] Shabnam Hassani. Replication Package: Evaluation Results, 2025. https://github.com/shabnamhasani/GherkinSpec/tree/main/Evaluation [Last accessed: May. 2025].

[107] Shabnam Hassani, Mehrdad Sabetzadeh, and Daniel Amyot. An empirical study on LLM-based classification of requirements-related provisions in food-safety regulations. *Empirical Software Engineering*, 30:article 72, 2025. doi:10.1007/s10664-025-10619-z.

[108] Shabnam Hassani, Mehrdad Sabetzadeh, and Daniel Amyot. From law to Gherkin: A human-centred quasi-experiment on the quality of LLM-generated behavioural specifications from food-safety regulations, 2025. arXiv:2508.20744.

[109] Shabnam Hassani, Mehrdad Sabetzadeh, Daniel Amyot, and Jain Liao. Rethinking legal compliance automation: Opportunities with large language models. In *Proceedings of 32nd IEEE International Requirements Engineering Conference (RE)*, pages 432–440. IEEE, 2024. doi:10.1109/RE59067.2024.00051.

[110] Petra Heck and Andy Zaidman. A quality framework for agile requirements: a practitioner's perspective, 2014. arXiv:1406.4692, doi:10.48550/arXiv.1406.4692.

[111] Aslak Hellesoy. *The Cucumber book: behaviour-driven development for testers and developers*. The Pragmatic Bookshelf, 2017. https://cucumber.io/docs/gherkin/reference/.

[112] Tobias Hey, Jan Keim, Anne Koziolek, and Walter F. Tichy. NoRBERT: Transfer learning for requirements classification. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pages 169–179. IEEE CS, 2020. doi:10.1109/RE48521.2020.00028.

[113] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi:10.1162/neco.1997.9.8.1735.

[114] Chris Jay Hoofnagle, Bart van der Sloot, and Frederik Zuiderveen Borgesius. The European Union General Data Protection Regulation: What It Is and What It Means. *Information & Communications Technology Law*, 28(1):65–98, 2019. doi:10.1080/13600834.2019.1573501.

[115] Hugging Face. Chatcompletion. https://bit.ly/48cuHrV.

[116] Hugging Face. Huggingface, 2016. https://huggingface.co/ [last accessed: Dec. 2024].

[117] Hugging Face. Llama-2-13b, 2023. https://bit.ly/3uvUo92.

[118] Hugging Face. Llama-2-7b, 2023. https://bit.ly/48fm3cf.

[119] Hugging Face. Mistral, 2023. https://bit.ly/49mJdOO.

[120] Hugging Face. MistralInstruct, 2023. https://bit.ly/3OBoXRB.

[121] Hugging Face. Mixtral, 2023. https://huggingface.co/mistralai.

[122] Hugging Face. Phi-2, 2023. https://bit.ly/3UAcpxu.

[123] Hugging Face. Zephyr, 2023. https://bit.ly/49vg3wP.

[124] Hugging Face. Llama, 2024. https://huggingface.co/meta-llama.

[125] Hugging Face. PEFT, 2025. https://huggingface.co/docs/peft/index.

[126] IBM. Openpages regulatory compliance management. https://www.ibm.com/products/openpages/regulatory-compliance.

[127] Muhammad Ilyas Azeem and Sallam Abualhaija. A multi-solution study on GDPR AI-enabled completeness checking of DPAs. *Empirical Software Engineering*, 29, 2023. doi:10.1007/s10664-024-10491-3.

[128] Chirag Jain, Preethu Rose Anish, Amrita Singh, and Smita Ghaisas. A transformer-based approach for abstractive summarization of requirements from obligations in software engineering contracts. In *2023 IEEE 31st International Requirements Engineering Conference (RE)*, pages 169–179. IEEE CS, 2023. doi:10.1109/RE57278.2023.00025.

[129] Susan Jamieson. Likert scales: How to (ab) use them? *Medical education*, 38(12):1217–1218, 2004. doi:10.1111/j.1365-2929.2004.02012.x.

[130] JBehave. JBehave official website, 2022. https://jbehave.org/.

[131] Andreas Jedlitschka, Marcus Ciolkowski, and Dietmar Pfahl. *Reporting Experiments in Software Engineering*, chapter 8 of Guide to Advanced Empirical Software Engineering, pages 201–228. Springer, 2008. doi:10.1007/978-1-84800-044-5_8.

[132] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. Mistral 7B, 2023. arXiv:2310.06825.

[133] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, et al. Mixtral of experts, 2024. arXiv:2401.04088.

[134] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Machine Learning: ECML-98*, pages 137–142. Springer, 1998.

[135] Andrew JI Jones and Marek Sergot. Deontic logic in the representation of law: Towards a methodology. *Artificial Intelligence and Law*, 1:45–64, 1992. doi:10.1007/BF00118478.

[136] Shanthi Karpurapu, Sravanthy Myneni, Unnati Nettur, Likhit Sagar Gajja, Dave Burke, Tom Stiehm, and Jeffery Payne. Comprehensive evaluation and insights into the use of large language models in the automation of behavior-driven development acceptance test formulation. *IEEE Access*, 12:58715–58721, 2024. doi:10.1109/ACCESS.2024.3391815.

[137] Evelyn Kempe, Samin Semsar, Aaron Massey, Sreedevi Sampath, and Carolyn Seaman. Modeling, analyzing and communicating regulatory ambiguity: An empirical study. In *2024 IEEE/ACM Workshop on Multi-disciplinary, Open, and RElevant Requirements Engineering (MO2RE)*, pages 28–34. ACM, 2024. doi:10.1145/3643666.3648576.

[138] Aniket Kesari, Travis Breaux, Tom Norton, Sarah Santos, and Anmol Singhal. From legal text to tech specs: Generative ai's interpretation of consent in privacy law, 2025. arXiv:2507.04185.

[139] Nadzeya Kiyavitskaya, Nicola Zeni, Travis D Breaux, Annie I Antón, James R Cordy, Luisa Mich, and John Mylopoulos. Automating the extraction of rights and obligations for regulatory compliance. In *Proceedings of 27th International Conference on Conceptual Modeling (ER)*, pages 154–168. Springer, 2008. doi:10.1007/978-3-540-87877-3_13.

[140] Aobo Kong et al. Better zero-shot reasoning with role-play prompting. In *NAACL-HLT'24*, volume 1, 2024. doi:10.18653/v1/2024.naacl-long.228.

[141] Oleksandr Kosenkov, Parisa Elahidoost, Tony Gorschek, Jannik Fischbach, Daniel Mendez, Michael Unterkalmsteiner, Davide Fucci, and Rahul Mohanani. Systematic mapping study on requirements engineering for regulatory compliance of software systems. *Information and Software Technology*, 178:107622, 2025. doi:10.1016/j.infsof.2024.107622.

[142] Oleksandr Kosenkov, Michael Unterkalmsteiner, Daniel Mendez, Davide Fucci, Tony Gorschek, and Jannik Fischbach. On developing an artifact-based approach to regulatory requirements engineering. In *2024 IEEE 32nd International Requirements Engineering Conference Workshops (REW)*, pages 262–271. IEEE CS, 2024. doi:10.1109/REW61692.2024.00041.

[143] Mohammed Lafi, Thamer Alrawashed, and Ahmad Munir Hammad. Automated test cases generation from requirements specification. In *2021 International Conference on Information Technology (ICIT)*, pages 852–857. IEEE CS, 2021. doi:10.1109/ICIT52682.2021.9491761.

[144] Lina Lam. Llama 3.3 just dropped — is it better than GPT-4 or Claude-Sonnet-3.5?, 2024. https://www.helicone.ai/blog/meta-llama-3-3-70-b-instruct.

[145] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language

representations. In *Proceedings of 8th International Conference on Learning Representations (ICLR)*, 2020. doi:10.48550/arXiv.1909.11942.

[146] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9), 2023. doi:10.1145/3560815.

[147] 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*, pages 2154–2164, 2021. doi:10.1145/3442381.3450022.

[148] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of 7th International Conference on Learning Representations (ICLR)*, 2019. doi:10.48550/arXiv.1711.05101.

[149] Garm Lucassen, Fabiano Dalpiaz, Jan Martijn E.M. van der Werf, and Sjaak Brinkkemper. Forging high-quality user stories: Towards a discipline for agile requirements. In *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, pages 126–135. IEEE CS, 2015. doi:10.1109/RE.2015.7320415.

[150] Garm Lucassen, Fabiano Dalpiaz, Jan Martijn EM van der Werf, and Sjaak Brinkkemper. Improving agile requirements: the quality user story framework and tool. *Requirements Engineering*, 21:383–403, 2016. doi:10.1007/s00766-016-0250-x.

[151] Clara Marie Luders, Tim Pietz, and Walid Maalej. Automated detection of typed links in issue trackers. In *Proceedings of 30th IEEE International Requirements Engineering Conference (RE)*, pages 26–38. IEEE, 2022. doi:10.48550/arXiv.2206.07182.

[152] Dipeeka Luitel, Shabnam Hassani, and Mehrdad Sabetzadeh. Using language models for enhancing the completeness of natural-language requirements. In *Proceedings of 29th International Working Conference on Requirement Engineering: Foundation for Software Quality (REFSQ)*, pages 87–104. Springer, 2023. doi:10.48550/arXiv.2302.04792.

[153] Dipeeka Luitel, Shabnam Hassani, and Mehrdad Sabetzadeh. Improving requirements completeness: Automated assistance through large language models. *Requirements Engineering*, 29(1):73–95, 2024. doi:10.1007/s00766-024-00416-3.

[154] Rainer Lutze and Klemens Waldhör. Generating specifications from requirements documents for smart devices using large language models (LLMs). In *Human-Computer Interaction*, pages 94–108. Springer, 2024. doi:10.1007/978-3-031-60405-8_7.

[155] Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D Manning, and Daniel E Ho. Hallucination-free? assessing the reliability of leading ai legal research tools. *Journal of Empirical Legal Studies*, 22(2):216–242, 2025. doi:10.1111/jels.12413.

[156] Manning publications. Writing Great Specifications, 2024. https://livebook.manning.com/book/writing-great-specifications/chapter-2/49 [last accessed: Dec. 2024].

[157] Milan Markovic, Peter Edwards, Martin Kollingbaum, and Alan Rowe. Modelling provenance of sensor data for food safety compliance checking. In *Proceedings of 6th International Provenance and Annotation Workshop (IPAW)*, page 134–145. Springer, 2016. doi:10.1007/978-3-319-40593-3_11.

[158] Nuno Marques, Rodrigo Rocha Silva, and Jorge Bernardino. Using ChatGPT in software requirements engineering: A comprehensive review. *Future Internet*, 16(6), 2024. doi:10.3390/fi16060180.

[159] Alok Mathur, Shreyaan Pradhan, Prasoon Soni, Dhruvil Patel, and Rajeshkannan Regunathan. Automated test case generation using T5 and GPT-3. In *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 1986–1992. IEEE, 2023. doi:10.1109/ICACCS57279.2023.10112971.

[160] Şevval Mehder and Fatma Başak Aydemir. Classification of issue discussions in open source projects using deep language models. In *2022 IEEE 30th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2022. doi:10.1109/REW56159.2022.00040.

[161] Rohan Reddy Mekala, Asif Irfan, Eduard C Groen, Adam Porter, and Mikael Lindvall. Classifying user requirements from online feedback in small dataset environments using deep learning. In *Proceedings of 29th IEEE International Requirements Engineering Conference (RE)*. IEEE, 2021. doi:10.1109/RE51729.2021.00020.

[162] Douglas C Montgomery. *Design and analysis of experiments*. John wiley & sons, 2017.

[163] Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12284–12314. Association for Computational Linguistics, 2023. doi:10.18653/v1/2023.findings-acl.779.

[164] Sara Moshtari, Ahmet Okutan, and Mehdi Mirakhorli. A grounded theory based approach to characterize software attack surfaces. In *Proceedings of 44th International Conference on Software Engineering (ICSE)*, pages 13–24. IEEE, 2022. doi:DOI:10.48550/arXiv.2112.01635.

[165] Najmeh Mousavi Nejad, Pablo Jabat, Rostislav Nedelchev, Simon Scerri, and Damien Graux. Establishing a strong baseline for privacy policy classification. *ICT Systems Security and Privacy Protection*, 580:370 – 383, 2020. doi:10.1007/978-3-030-58201-2_25.

[166] Jakob Mund, Daniel Mendez Fernandez, Henning Femmer, and Jonas Eckhardt. Does quality of requirements specifications matter? combined results of two empirical studies. In *ESEM'15*, 2015. doi:10.1109/ESEM.2015.7321195.

[167] Anmol Nayak, Hari Prasad Timmapathini, Vidhya Murali, Karthikeyan Ponnalagu, Vijendran Gopalan Venkoparao, and Amalinda Post. Req2Spec: Transforming software requirements into formal specifications using Natural Language Processing. In *Requirements Engineering: Foundation for Software Quality*, pages 87–95. Springer, 2022. doi:10.1007/978-3-030-98464-9_8.

[168] Gabriel Oliveira and Sabrina Marczak. On the empirical evaluation of BDD scenarios quality: Preliminary findings of an empirical study. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 299–302. IEEE CS, 2017. doi:10.1109/REW.2017.62.

[169] Gabriel Oliveira, Sabrina Marczak, and Cassiano Moralles. How to evaluate BDD scenarios' quality? In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, SBES '19, pages 481–490. ACM, 2019. doi:10.1145/3350768.3351301.

[170] OneTrust. Ai governance. https://www.onetrust.com/solutions/ai-governance/.

[171] Open AI. Temperaturerandomness, 2023. https://bit.ly/TempRandomness.

[172] Open Policy Agent. Open policy agent (opa). https://openpolicyagent.org/.

[173] OpenAI. Openai, 2018. https://openai.com/ [last accessed: Dec. 2024].

[174] OpenAI. GPT-4 technical report. *arXiv:2303.08774*, 2023.

[175] OpenAI. OpenAI – fine-tuning, 2024. https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset [last accessed: Dec. 2024].

[176] OpenAI. Best practices for prompt engineering with the OpenAI API, 2025. https://bit.ly/3LVJRfT.

[177] Paul N. Otto and Annie I. Antón. Addressing legal requirements in requirements engineering. In *15th IEEE International Requirements Engineering Conference (RE 2007)*, pages 5–14. IEEE CS, 2007. doi:10.1109/RE.2007.65.

[178] Palantir Technologies Inc. Best practices for LLM prompt engineering, 2025. https://bit.ly/4pCv6gl.

[179] Panda BDD page. Behavioral Driven Development, 2024. https://automationpanda.com/bdd/ [last accessed: Dec. 2024].

[180] PCI Security Standards Council. Payment card industry (PCI) data security standard: Requirements and security assessment procedures. https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2-1.pdf, 2018. Version 3.2.1.

[181] Simon Pearson, Steve Brewer, Phil Godsiff, and Roger Maull. Food Data Trust: A framework for information sharing. *Food Stand. Agency*, pages 543–545, 2021. doi:10.1038/s43016-021-00346-1.

[182] Simon Pearson, Steve Brewer, Louise Mannings, Luc Dosoudilova, George Onoufriou, Aiden Durrant, Georgios Leontidis, Charbel Jabbarpour, Andrea Zisman, Gerard Parr, et al. Decarbonising our food systems: contextualising digitalisation for net zero. *Frontiers in Sustainable Food Systems*, 7, 2023. doi:10.3389/fsufs.2023.1049229.

[183] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, 2014. doi:10.3115/v1/D14-1162.

[184] Hung Viet Pham, Shangshu Qian, Jiannan Wang, Thibaud Lutellier, Jonathan Rosenthal, Lin Tan, Yaoliang Yu, and Nachiappan Nagappan. Problems and opportunities in training deep learning software systems: An analysis of variance. In *Proceedings of 35th IEEE/ACM international conference on automated software engineering (ASE)*, pages 771–783, 2020. doi:10.1145/3324884.3416545.

[185] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training, 2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[186] Tajmilur Rahman and Yuecai Zhu. Automated user story generation with test case specification using large language model, 2024. doi:10.48550/arXiv.2404.01558.

[187] Nils Reimers. Encoders, 2023. https://bit.ly/49wjJ1r.

[188] Nils Reimers and Iryna Gurevych. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017. URL: https://aclanthology.org/D17-1035/, doi:10.18653/v1/D17-1035.

[189] Research and Markets. Food Safety Testing - A Global Market Overview, 2024. https://bit.ly/3YohZ7C [last accessed: Dec. 2024].

[190] Krishna Ronanki, Christian Berger, and Jennifer Horkoff. Investigating ChatGPT's potential to assist in requirements elicitation processes. In *2023 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 354–361. IEEE CS, 2023. doi:10.1109/SEAA60479.2023.00061.

[191] Abhishek Sainani, Preethu Rose Anish, Vivek Joshi, and Smita Ghaisas. Extracting and classifying requirements from software engineering contracts. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pages 147–157. IEEE CS, 2020. doi:10.1109/RE48521.2020.00026.

[192] Johnny M Saldaña. *The coding manual for qualitative researchers*. SAGE Publications, 2015. URL: https://collegepublishing.sagepub.com/products/the-coding-manual-for-qualitative-researchers-4-273583.

[193] Joana Salgueiro, Fernando Ribeiro, and José Metrôlho. Best practices for outsystems development and its influence on test automation. In *Trends and Applications in Information Systems and Technologies*, pages 85–95. Springer, 2021. doi:10.1007/978-3-030-72654-6_9.

[194] Nicolas Sannier, Morayo Adedjouma, Mehrdad Sabetzadeh, and Lionel Briand. An automated framework for detection and resolution of cross references in legal texts. *Requirements Engineering*, 22:215–237, 2017. doi:10.1007/s00766-015-0241-3.

[195] Sarah Santos, Travis Breaux, Thomas Norton, Sara Haghighi, and Sepideh Ghanavati. Requirements satisfiability with in-context learning. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 168–179. IEEE CS, 2024. doi:10.1109/RE59067.2024.00025.

[196] Sarah Santos, Travis Breaux, Thomas Norton, Sara Haghighi, and Sepideh Ghanavati. Requirements satisfiability with in-context learning. In *32nd IEEE International Requirements Engineering Conference (RE'24)*, pages 168–179, 2024. doi:10.1109/RE59067.2024.00025.

[197] scikit-learn. Random forest random_state, 2024. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html [last accessed: Dec. 2024].

[198] ServiceNow. Governance, risk, and compliance (grc). https://www.servicenow.com/products/governance-risk-and-compliance.html.

[199] Jianwei Shi, Alan Dryaev, and Kurt Schneider. Using an AI chatbot to refine Gherkin specifications based on stakeholder comments. Technical report, Hannover: Institutionelles Repositorium der Leibniz Universität, 2024. doi:10.15488/17361.

[200] Anmol Singhal and Travis Breaux. Legal requirements translation from law, 2025. arXiv:2507.02846.

[201] Amin Sleimi, Marcello Ceci, Nicolas Sannier, Mehrdad Sabetzadeh, Lionel Briand, and John Dann. A query system for extracting requirements-related information from legal texts. In *Proceedings of 27th IEEE International Requirements Engineering Conference (RE)*, pages 319–329. IEEE, 2019. doi:10.1109/RE.2019.00041.

[202] Amin Sleimi, Nicolas Sannier, Mehrdad Sabetzadeh, Lionel Briand, and John Dann. Automated extraction of semantic legal metadata using natural language processing. In *Proceedings of 26th IEEE International Requirements Engineering Conference (RE)*, pages 124–135. IEEE, 2018. doi:10.1109/RE.2018.00022.

[203] John Ferguson Smart and Jan Molak. *BDD in Action: Behavior-Driven Development for the whole software lifecycle*. Simon and Schuster, 2023. URL: https://books.google.ca/books/about/BDD_in_Action_Second_Edition.html?id=hIK3EAAAQBAJ.

[204] Mathias Soeken, Robert Wille, and Rolf Drechsler. Assisted Behavior Driven Development using Natural Language Processing. In *Objects, Models, Components, Patterns*, pages 269–287. Springer, 2012. doi:10.1007/978-3-642-30561-0_19.

[205] Hiago Natan Fernandes Sousa et al. Um experimento comparativo da eficácia de diferentes LLM na geração de cenários gherkin. Master's thesis, Universidade Federal de Campina Grande, Brasil, 2025. URL: http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/41048.

[206] Paola Spoletini and Alessio Ferrari. The return of formal requirements engineering in the era of Large Language Models. In *Requirements Engineering: Foundation for Software Quality*, pages 344–353. Springer, 2024. doi:10.1007/978-3-031-57327-9_22.

[207] Klaas-Jan Stol, Paul Ralph, and Brian Fitzgerald. Grounded theory in software engineering research: A critical review and guidelines. In *Proceedings of 38th International Conference on Software Engineering (ICSE)*, pages 120–131, 2016. doi:10.1145/2884781.2884833.

[208] Anselm Strauss and Juliet Corbin. *Basics of Qualitative Research – Techniques and Procedures for Developing Grounded Theory (2nd Ed.)*. SAGE publications, 1998.

[209] Simeng Sun, Yang Liu, Shuohang Wang, Dan Iter, Chenguang Zhu, and Mohit Iyyer. PEARL: Prompting large language models to plan and execute actions over long documents. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 469–486. Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.eacl-long.29.

[210] Chenhao Tang, Zhengliang Liu, Chong Ma, Zihao Wu, Yiwei Li, Wei Liu, Dajiang Zhu, Quanzheng Li, Xiang Li, Tianming Liu, et al. PolicyGPT: Automated analysis of privacy policies with large language models, 2023. doi:10.48550/arXiv.2309.10238.

[211] TechTalk. Specflow, 2025. Open-source BDD automation for .NET. URL: https://specflow.org.

[212] The Guardian. Lawyer caught using AI-generated false citations in court case penalised in australian first, 2025. https://bit.ly/4pkgDWR.

[213] 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. IEEE CS, 2020. doi:10.1109/RE48521.2020.00025.

[214] Damiano Torre, Mauricio Alferez, Ghanem Soltana, Mehrdad Sabetzadeh, and Lionel Briand. Modeling data protection and privacy: application and experience with GDPR. *Software and Systems Modeling*, 20(6):2071–2087, 2021. doi:10.1007/s10270-021-00935-5.

[215] Damiano Torre, Ghanem Soltana, Mehrdad Sabetzadeh, Lionel C Briand, Yuri Auffinger, and Peter Goes. Using models to enable compliance checking against the GDPR: an experience report. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pages 1–11, 2019. doi:10.1109/MODELS.2019.00-20.

[216] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models, 2023. arXiv:2302.13971.

[217] U.S. Congress. Health insurance portability and accountability act of 1996. https://www.congress.gov/bill/104th-congress/house-bill/3103, 1996.

[218] U.S. Food & Drug Administration. Packaging for foods treated with ionizing radiation, 2018. https://www.fda.gov/food/food-ingredients-packaging/irradiation-food-packaging [Last accessed: Dec. 2024].

[219] U.S. Food & Drug Administration. Guidance for industry: Preparation of food contact notifications for food contact substances in contact with infant formula and/or human milk, 2019. https://bit.ly/GuidanceforIndustry [Last accessed: Dec. 2024].

[220] U.S. Food & Drug Administration. Egg safety: Final rule, 2022. https://bit.ly/FDAEggSafety [Last accessed: Dec. 2024].

[221] U.S. Food & Drug Administration. Food labeling: Nutrient content claims, definition of term "healthy", 2022. https://bit.ly/FDANutrients [Last accessed: Dec. 2024].

[222] Vasily Varenov and Aydar Gabdrahmanov. Security requirements classification into groups using NLP transformers. In *Proceedings of 29th IEEE International Requirements Engineering Conference Workshops (REW)*. IEEE, 2021. doi:10.1109/REW53955.2021.9714713.

[223] Andras Vargha and Harold Delaney. A critique and improvement of the CL common language effect size statistics of McGraw and Wong. *Journal of Educational and Behavioral Statistics*, 25(2):101–132, 2000. doi:10.3102/10769986025002101.

[224] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017. doi:10.5555/3295222.3295349.

[225] Bill Wake. INVEST in Good Stories, and SMART Tasks, 2003. https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/.

[226] Fernando Wanderley, Antonio Silva, and João Araújo. Evaluation of BehaviorMap: A user-centered behavior language. In *2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*, pages 309–320. IEEE CS, 2015. doi:10.1109/RCIS.2015.7128891.

[227] Fanyu Wang, Chetan Arora, Yonghui Liu, Kaicheng Huang, Chakkrit Tantithamthavorn, Aldeida Aleti, Dishan Sambathkumar, and David Lo. Multi-modal requirements data-based acceptance criteria generation using LLMs. *arXiv:2508.06888*, 2025. arXiv:2508.06888, doi:10.48550/arXiv.2508.06888.

[228] Yawen Wang, Lin Shi, Mingyang Li, Qing Wang, and Yun Yang. A deep context-wise method for coreference detection in natural language requirements. In *Proceedings of 28th IEEE International Requirements Engineering Conference (RE)*, pages 180–191. IEEE, 2020. doi:10.1109/RE48521.2020.00029.

[229] Yawen Wang, Lin Shi, Mingyang Li, Qing Wang, and Yun Yang. Detecting coreferent entities in natural language requirements. *Requirements Engineering*, 27(3):351–373, 2022. doi:10.1007/s00766-022-00374-8.

[230] Yves Wautelet, Anousheh Khajeh Nassiri, and Konstantinos Tsilionis. Investigating quality attributes in Behavior-Driven Development scenarios: An evaluation framework and an experimental supporting tool. In *The Practice of Enterprise Modeling*, pages 125–142. Springer, 2024. doi:10.1007/978-3-031-48583-1_8.

[231] Bingyang Wei. Requirements are all you need: From requirements to code with LLMs. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 416–422. IEEE CS, 2024. doi:10.1109/RE59067.2024.00049.

[232] Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N Cameron Russell, et al. The creation and analysis of a website privacy policy corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1330–1340. Association for Computational Linguistics, 2016. doi:10.18653/v1/P16-1126.

[233] Katharina Winter, Henning Femmer, and Andreas Vogelsang. How do quantifiers affect the quality of requirements? In *REFSQ'20*, 2020. doi:10.1007/978-3-030-44429-7_1.

[234] World Health Organization. Food safety, 2024. https://www.who.int/news-room/fact-sheets/detail/food-safety [Last accessed: April. 2025].

[235] Ning Wu, Ming Gong, Linjun Shou, Shining Liang, and Daxin Jiang. Large language models are diverse role-players for summarization evaluation. In *NLPCC'23*, 2023. doi:10.1007/978-3-031-44693-1_54.

[236] Anhao Xiang, Weiping Pei, and Chuan Yue. PolicyChecker: Analyzing the GDPR completeness of mobile apps' privacy policies. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, page 3373–3387. Association for Computing Machinery, 2023. doi:10.1145/3576915.3623067.

[237] Danning Xie, Byungwoo Yoo, Nan Jiang, Mijung Kim, Lin Tan, Xiangyu Zhang, and Judy S. Lee. How effective are large language models in generating software specifications?, 2025. URL: https://arxiv.org/abs/2306.03324, arXiv:2306.03324, doi:10.48550/arXiv.2306.03324.

[238] Fuman Xie, Yanjun Zhang, Chuan Yan, Suwan Li, Lei Bu, Kai Chen, Zi Huang, and Guangdong Bai. Scrutinizing privacy policy compliance of virtual personal assistant apps. In *Proceedings of 37th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. Association for Computing Machinery, 2022. doi:10.1145/3551349.3560416.

[239] N. Zeni, E. A. Seid, P. Engiel, S. Ingolfo, and J. Mylopoulos. Building large models of law with NómosT. In *Conceptual Modeling*, pages 233–247. Springer, 2016. doi:10.1007/978-3-319-46397-1_18.

[240] Nicola Zeni, Nadzeya Kiyavitskaya, Luisa Mich, James R Cordy, and John Mylopoulos. GaiusT: supporting the extraction of rights and obligations for regulatory compliance. *Requirements Engineering*, 20:1–22, 2015. doi:10.1007/s00766-013-0181-8.

[241] Liu Zhuang, Shi Wayne, Lin an Ya, and Zhao Jun. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227. Springer, 2021. doi:10.1007/978-3-030-84186-7_31.