CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

An NLP-based System for Automated Compliance Analysis and Requirement Classification in Engineering Applications

Master's Thesis in Computer science and engineering

Shuyue Ding Johan Lindén

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

Master's Thesis 2025

An NLP-based System for Automated Compliance Analysis and Requirement Classification in Engineering Applications

Shuyue Ding, Johan Lindén

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

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

An NLP-based System for Automated Compliance Analysis and Requirement Classification in Engineering Applications
Shuyue Ding, Johan Lindén

Supervisor: Irum Inayat, Department of Computer Science and Engineering
Examiner: Mazen Mohamad, Department of Computer Science and Engineering

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

Typeset in LaTeX
Gothenburg, Sweden 2025

Shuyue Ding,
Department of Physics
Chalmers University of Technology and University of Gothenburg

Johan Lindén
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Regulatory compliance is a critical challenge in engineering product development, particularly in industries governed by complex and frequently evolving standards. This research, conducted in collaboration with Volvo Penta, explores the use of Natural Language Processing (NLP) techniques to automate the classification and interpretation of regulatory clauses in support of early-stage requirements analysis, with a focus on ensuring traceability, usability, and transparency in the decision-making process. The proposed method integrates rule-based preprocessing, domain-specific keyword enrichment, semantic similarity retrieval using dense embeddings, and reasoning about individual regulatory statements using large language models (LLMs). The system is connected to Volvo Penta's SystemWeaver platform to support traceable, explainable, and human-reviewable predictions at the clause level. By automating the identification and interpretation of relevant regulatory content, the system reduces manual workload, improves consistency, and enables scalable compliance workflows. Evaluation through both quantitative analysis and expert feedback indicates strong alignment with human judgment and demonstrates the system's value as a decision-support tool in industrial engineering compliance processes. While limitations remain in handling edge cases and ambiguous regulatory language, results suggest that NLP-driven methods can meaningfully support scalable, traceable, and more efficient compliance processes in industrial engineering settings.

Keywords: Natural Language Processing, Regulatory Compliance, Engineering Requirements, Large Language Models, Semantic Embeddings, SystemWeaver, Human-in-the-Loop.

Acknowledgements

First and foremost, we would like to thank our team of supervisors, Irum Inayat, Magnus Liske, Patricija Svartborn, and our examiner Mazen Mohamad for going above and beyond what one might expect from a supervisor. Your guidance, availability, and expertise have been pivotal to the success of our thesis. We would also like to express our gratitude to Volvo Penta for making this project possible.

Shuyue Ding, Johan Lindén, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AI	Artificial Intelligence
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
CE	Conformité Européenne
CoT	Chain-of-Thought
CSV	Comma-Separated Values
EC	European Commission
EMC	Electromagnetic Compatibility
EM	Electromagnetic
EN	European Norm
EU	European Union
FAISS	Facebook AI Similarity Search
GDPR	General Data Protection Regulation
HITL	Human-in-the-Loop
HTML	HyperText Markup Language
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
JSON	JavaScript Object Notation
LLM	Large Language Model
ML	Machine Learning
NLP	Natural Language Processing
OEM	Original Equipment Manufacturer
RoHS	Restriction of Hazardous Substances
RS	Requirement Specification
TD	Target Description
XML	eXtensible Markup Language

Contents

List of Figures

List of Tables

1

Introduction

1.1 Problem Description

Product compliance is the decisive gateway between an engineering product and its target market, governing whether a product can legally be placed on shelves, installed in the field, or exported across borders [1]. A single misinterpretation of a clause in, for example, the European Union (EU) Machinery Regulation or the United States (US) Environmental Protection Agency (EPA) rules can expose a company to multimillion-euro fines, trigger recalls, or force the scrapping and rework of entire production lots, resulting in substantial inventory write-offs and downstream schedule delay [2][3]. Compliance spans a range of regulatory pillars, including electrical and mechanical safety, electromagnetic compatibility, hazardous substances restrictions (e.g. Restriction of Hazardous Substances (RoHS), the Regulation on the Registration, Evaluation, Authorisation and restriction of Chemicals (REACH), energy efficiency, and emerging cybersecurity mandates. For example, the EU RoHS and REACH directives restrict hazardous materials to protect human health and the environment [4], the Electromagnetic Compatibility (EMC) Directive ensures electromagnetic compatibility in electrical products [5], and vehicle regulations such as United Nations Regulation No. 155 (UN R155) impose mandatory cybersecurity requirements on connected systems [6].

Because every sales region layers its own directives, harmonised standards, and guidance documents on top of international frameworks (International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Underwriters Laboratories (UL), etc.), engineers must routinely screen hundreds of heterogeneous documents to locate and interpret the handful of clauses that actually constrain a given product variant [7]. These documents are expressed in free-form natural language, often with nested lists, exceptions, and cross-references, making the relevant information difficult to extract systematically [8]. Subject matter experts therefore spend hundreds of hours reading line by line, extracting obligations and mapping them to bill-of-material items or evidential links for auditors. The volume of regulation continues to rise, update cycles are shortening, and the same manufacturer may now ship dozens of variants into dozens of jurisdictions [9]. Manual review is becoming unsustainable, and the industry increasingly recognises the need for a systematic, automated approach to regulatory interpretation [10].

Natural language processing (NLP) provides a technological foundation for such automation. Classical techniques, such as part-of-speech (POS) tagging, named entity recognition, and dependency parsing, have already been applied to information extraction in legal and architectural domains [11, 12]. Transformer-based language models such as Bidirectional Encoder Representations from Transformers (BERT) provide dense semantic embeddings that allow for high precision similarity matching and clustering of conceptually related clauses [13]. Together, these tools support compliance orientated tasks such as clause classification, requirement mining, and consistency checking. However, a critical gap remains: Models trained in homogeneous legal, medical, or software corpora transfer poorly to engineering standards because vocabulary, document structure, and contextual signals differ significantly across domains. Engineering texts mention specialised components ("charge-air cooler", 'pressure relief valve'), physical units, and domain-specific relationships that generic NLP pipelines do not capture out of the box. Consequently, published compliance automation studies have focused on software licence texts, privacy regulations, and clinical guidelines, leaving the engineering field largely underexplored [14]. To address this gap, this thesis investigates the application of domain-adapted NLP techniques for engineering regulatory compliance. The aim is to explore scalable methods for transforming unstructured regulatory text into actionable product requirements suitable for traceable and verifiable workflows.

1.2 Purpose of the Study

This thesis is conducted in collaboration with Volvo Penta, a business area within the Volvo Group that develops engines and power systems for marine and industrial applications. Volvo Penta currently relies on SystemWeaver, an internal platform that stores regulatory materials and links them to specific products and projects. Although SystemWeaver provides organised access to these documents, the task of translating those standards into clear, actionable requirements remains mostly manual. For each regulatory document, a human expert must review the text, assess its relevance, and formulate a conclusion that helps guide the engineering team. These conclusions can be explicit requirements such as 'The battery voltage must remain below 80V' or statements indicating that a particular standard does not apply. Although SystemWeaver enables traceability and documentation, it does not perform automated regulatory analysis. As a result, engineers are tasked with manually interpreting regulatory material, frequently needing to reproduce or rephrase standard sections to extract guidance specific to the product. This process is labour-intensive and prone to errors.

This research proposes one method for automated regulatory breakdown and interpretation focused on the engineering field, which can provide valuable reference for product developers and accelerate the product development process. In addition, considering industrial applications, this study aims to evaluate the trade-off between interpretability and accuracy between different methods. The method proposed in this study utilises techniques such as similarity computation and keyword matching to retrieve standard terms applied to specific projects. After that, large language

model methods are utilised to generate structured and executable conclusions. This project also developed a complete set of end-to-end tools that can be embedded in existing compliance workflows and integrated with the Hybrid Human-in-the-Loop approach. The entire approach can improve work efficiency and provide a valuable reference for domain experts.

Research Questions

In this thesis, we aim to answer the following three research questions:

- **RQ1:** *How can we develop a system that automatically links regulatory requirements to relevant product and engineering information in Volvo Penta's lifecycle?*
- **RQ2:** *How can this system be effectively integrated into Volvo Penta's engineering workflows to support traceability, usability, and transparency in decision-making?*
- **RQ3:** *How does the system perform in terms of accuracy, interpretability, and explainability, and to what extent does it support decision-making based on expert evaluations and real-world scenarios?*

RQ1 focuses on the technical development of the system. It addresses how to represent regulatory and engineering data, and how to design a model that accurately links regulatory requirements to relevant product features, specifications, and artifacts. This includes selecting the appropriate embedding methods, handling the domain-specific language, and ensuring coverage across multiple regulation formats. The outcome should enable a foundation for traceable and interpretable mappings that support downstream transparency needs.

RQ2 investigates how the developed system can be integrated into the existing Volvo Penta engineering workflows. The goal is to ensure that the system aligns with current practices, supports traceability, and improves operational compliance, while also providing transparency in how links are generated. This includes user interface design, integration into tools such as SystemWeaver, and human-in-the-loop mechanisms that allow engineers to inspect, verify, and correct automated outputs. Here, explainability is treated as a usability requirement that enables trust and oversight during integration.

RQ3 evaluates the system from both a technical and user-centred perspective. Specifically, the question explores how well the system performs in terms of accuracy, interpretability, and explainability, and assesses its usefulness for decision support in real engineering scenarios. The evaluation combines expert interviews, industrial case studies, and quantitative validation metrics such as precision, recall, and confidence calibration. Special attention is paid to how clearly the system communicates its reasoning and how easily users can trace and validate its decisions.

These insights help to determine the readiness of the system for deployment and its value to compliance engineers.

1.3 Significance of the Study

The significance of this study lies in its contribution to both academic research and industrial practice. For the academic community, it advances the use of natural language processing in requirements engineering by applying and evaluating semantic and generative techniques in the context of engineering compliance, an area that remains underexplored. For industry, the study addresses a concrete need at Volvo Penta to reduce the manual burden of regulatory interpretation. The proposed approach may serve as a foundation for future compliance support tools and can be adapted by other organisations facing similar challenges.

2

Background

This chapter outlines the industrial and information management context in which the thesis is carried out. It begins by describing the general requirements landscape and the types of regulatory documents relevant to product development. This is followed by an overview of the Volvo Penta phase-gate model and the SystemWeaver repository, which stores artefacts later used in the automation approach. The technical background related to the proposed method, such as embeddings and LLMs, is then introduced. The chapter concludes with a terminology section that defines key concepts such as *transparency*, *usability*, and *explainability*, which are central to the system design and evaluation.

2.1 Regulations and the Spectrum of Standards

In EU law, a **regulation** is a binding legislative act that must be applied in its entirety in all Member States. The regulations are directly applicable and do not require national transposition. For example, the EU Machinery Regulation (EU) 2023/1230 applies uniformly across the EU upon its entry into force [15].

In contrast, a **directive** is a legislative act that sets a goal that all EU countries must achieve. However, it allows Member States the flexibility to devise their own laws on how to reach these goals. An example is the Low Voltage Directive 2014/35/EU, which mandates safety objectives for electrical equipment but allows national authorities to determine the specific implementation measures [15].

Formal Consensus Standards

A **standard** is an agreed-upon specification for a product, service, process, or interface. The issuing body and level of consensus can vary:

- **International standards**—published by organisations such as the ISO [16], the IEC [17] or the IMO [18]. An example is ISO 26262, which refers to functional safety in automotive systems [19].
- **Regional standards**—developed by European Standards Organisations (ESOs) such as the European Committee for Standardisation (CEN), the European Committee for Electrotechnical Standardisation (CENELEC) or the European

Telecommunications Standards Institute (ETSI). These are often adopted verbatim as national standards, such as the series 'EN' [20].

- **National standards**—issued by national standardisation bodies, for example, Deutsches Institut für Normung (DIN) in Germany [21], British Standards Institution (BSI) in the UK [22] and Swedish Standards Institute (SIS) in Sweden [23].

Harmonised Standards

Within the EU, certain European standards are designated as **harmonised standards**. These are developed by recognised ESOs following a request from the European Commission. Compliance with harmonised standards provides a presumption of conformity with the corresponding essential requirements of relevant EU legislation, facilitating the CE marking process [24].

A relevant example for this thesis is the **EN 1175:2020** standard, which specifies the electrical and electronic requirements for the design and construction of electrical installations in self-propelled industrial trucks. This standard addresses safety requirements for all electrical and electronic components, including safety-related parts of control systems, and is intended to be used in conjunction with EN ISO 3691-5:2015 [25].

Industry Standards

Certain industries are governed by *industry specifications* which, although not formal consensus standards, are widely adopted due to their relevance and utility. Examples include:

- **SAE J1939**—a communication protocol for vehicle networks, used mainly in heavy-duty vehicles. It defines standards for network communication among vehicle components [26].

- **AUTOSAR** (AUTomotive Open System Architecture) - a standardised automotive software architecture developed by a consortium of automotive manufacturers and suppliers to facilitate the development of automotive software [27].

Compliance with these specifications is often contractually or commercially mandated, although they are not formal consensus standards.

Company-Internal Standards and Operating Procedures

Large manufacturers maintain internal corporate or engineering standards that may reference public standards, but are tailored to their specific product ranges and operational contexts. For example, Volvo Group has its own set of corporate standards, such as STD 5019, which pertains to corrosion protection requirements for components and systems [28].

These internal standards ensure consistency, quality, and compliance across the organisation's products and processes, aligning with both external regulations and internal objectives.

Relevance to the Thesis

In the context of European product compliance, legal and technical requirements are layered and interconnected:

- **Regulations** are binding legislative acts that apply directly in all EU Member States without transposition (e.g., Machinery Regulation (EU) 2023/1230).
- **Directives** define required outcomes (e.g., EMC or Low-Voltage Directives) but must be implemented through national law.
- **Harmonised standards** (e.g., EN 1175) are consensus-based technical specifications that support compliance with directives. Products built in accordance with these standards are presumed to meet the essential safety or performance requirements of the associated legislation.
- **Legal requirements** originate from the directives and regulations themselves, but are often operationalised by adherence to relevant harmonised or international standards.

Together, this legal and normative structure determines whether a product can be legally placed on the market. Manufacturers must assess each clause across these layers to determine whether it constitutes a *requirement* and whether that requirement is *applicable* to a specific product configuration.

The automation prototype developed in this thesis addresses this compliance task. It focuses on a selected set of documents: the harmonised standard **EN 1175**, which is highly relevant for safety-related electrical systems in industrial vehicles, and three **EU directives** that do not include clause-level classifications. This mixed data set allows the system to be tested both retrospectively (against known labels) and prospectively (on unclassified material), thereby covering both verification accuracy and real-world usability.

From Legal Obligation to Engineering Requirement:

In practice, legal documents such as directives and harmonised standards are not merely compliance checklists, they are translated into concrete engineering requirements. Each relevant clause must be interpreted in terms of how it impacts the design, validation, and operation of a product. For this reason, the automation pipeline developed in this thesis treats each regulatory clause as a candidate requirement and assigns labels indicating its applicability. This framework aligns regulatory compliance with the standard lifecycle of requirements engineering [29].

2.2 Volvo Penta Phase-Gate Model

Volvo Penta's product development follows a structured phase-gate process, a widely adopted model in engineering project management. A phase–gate model (also known as a stage–gate model) is a well-known governance framework used to manage risk and ensure milestone readiness throughout the development lifecycle [30]. Each phase ends with a formal 'gate' where senior stakeholders evaluate the status of the project and decide whether it should proceed.

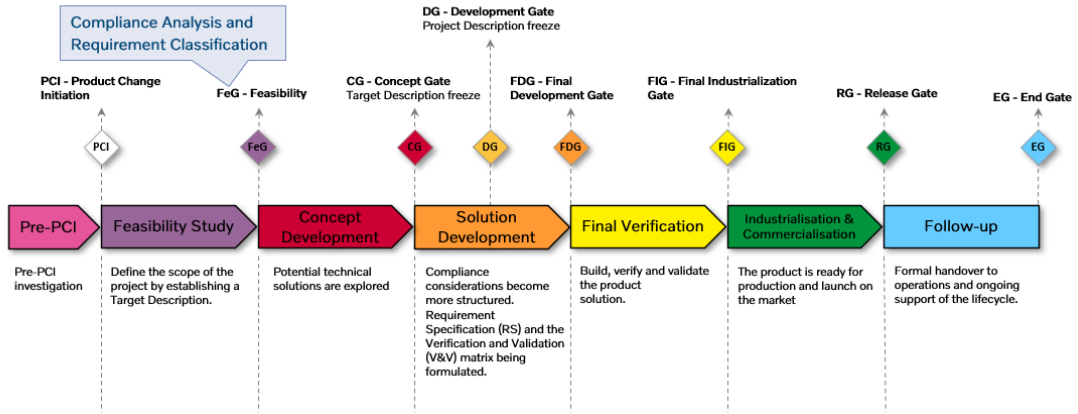

Figure 2.1: Volvo Penta phase-gate model and main deliverables.

Feasibility and Conceptualisation

The early stages begin with strategic planning and feasibility analysis. The **feasibility gate (FeG)** marks the point where initial business viability, technical alternatives, and scope of the programme are assessed and approved. Deliverables include a draft business case and an initial **Target Description (TD)**.

Concept Development

During the concept development phase, potential technical solutions are explored. A preferred concept is selected and captured in the Target Description, which is baselined at the **Concept Gate (CG)**. This document forms the basis for high-level system requirements.

Solution Development

The solution development phase refines the selected concept into a detailed system design. The **Development Gate (DG)** freezes the Project Description (PD), elaborating on the TD. During this phase, compliance considerations become more structured, with early drafts of the Requirement Specification (RS) and the Verification and Validation (V&V) matrix being formulated.

Final Verification

This phase involves system integration, testing, and formal validation. The **Final Development Gate (FDG)** represents a readiness check for industrialisation. By this point, the RS is fully defined and the V&V matrix must be executable and aligned with regulatory obligations.

Industrialisation and Release

In the final stages, the product is ready for production and launch on the market. The **Release Gate (RG)** ensures that compliance documentation and all supporting deliverables are complete. The **End Gate (EG)** signifies formal handover to operations and ongoing support of the lifecycle.

Compliance Across Gates

Throughout these phases, compliance evidence accumulates incrementally. Although this thesis focusses primarily on early stage analysis at the **FeG**, where the potential regulatory impact of new concepts must be assessed, the automated tool developed is designed for use throughout the process. It supports clause-level traceability from initial TD creation through to industrialisation and release, offering consistent, auditable, and updatable compliance assessments at each decision point.

2.3 Requirement-Document Landscape

Regulatory texts such as the Machinery Directive or harmonised standards such as EN 1175 are treated within Volvo Penta as structured sources of *external requirements*. These legal clauses are decomposed, interpreted, and incorporated into the same traceability model as internal functional or performance requirements. This integration ensures that legal obligations are embedded early in system architecture and verification planning. By treating legal clauses as first-class requirement items, the company can manage compliance as an integrated part of the product development lifecycle [31].

To support this, compliance and engineering requirements are governed through a structured set of configuration-controlled artefacts. These documents are maintained within SystemWeaver, a centralised product information management system used throughout the Volvo Group.

This section introduces the four core artefacts used to define, trace, and verify requirements throughout the product lifecycle. Among these, the **TD** is the main focus of this thesis, as early stage compliance screening, before FeG, offers the greatest leverage in identifying and mitigating regulatory risks at low cost. The relationships and information flow between these artefacts are summarised in Figure 2.2, which illustrates how requirements evolve from high level product goals to verifiable specifications.

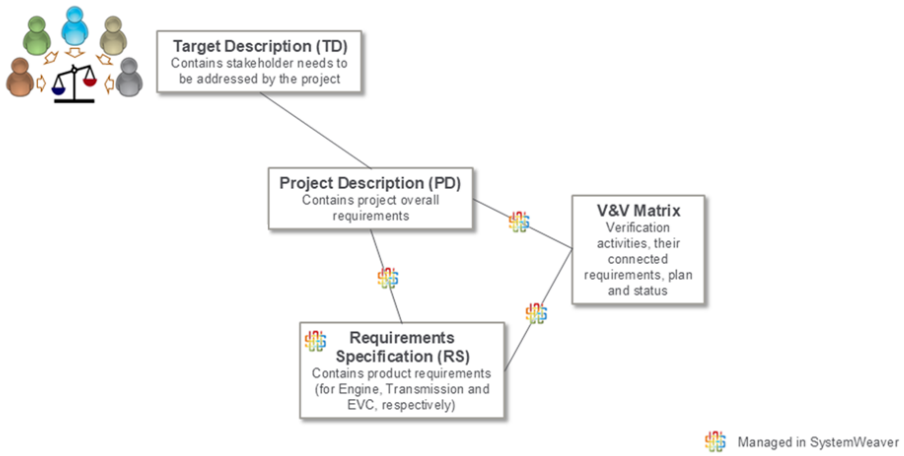

Figure 2.2: Information flow between requirement artefacts managed in SystemWeaver.

Target Description (TD)

The TD originates in the marketing and product planning organisation and serves as the earliest structured expression of a programme's scope. It outlines:

- business objectives and stakeholder needs.
- target markets, usage environments and operating conditions.
- high-level functional goals and technical constraints.

The TD is baselined at the **FeG** and provides the basis for the development of downstream requirements. Because it is both relatively stable and available early, the TD is an ideal input for automation. In this thesis, it is used to determine the relevance and applicability of individual clauses from regulatory documents such as EN 1175 and EU directives.

Other Core Artefacts

Although the TD is the primary input to the automation pipeline developed in this study, several other documents are essential to the end-to-end compliance process at Volvo Penta.

- **Project Description (PD):** Establishes the contractually agreed scope of the project, including budget, milestones, and deliverables. Baselined at the **Development Gate (DG)**.

- **Requirement Specification (RS):** Contains detailed structured engineering requirements organised by system and function. RS entries are traced to both the TD and the corresponding test cases in the V&V matrix.
- **Verification and Validation Matrix (V&V):** Defines how each requirement will be verified, by analysis, testing or inspection. The V&V matrix tracks test results and serves as the authoritative compliance audit trail during gate transitions.

Traceability from Requirements to Product Architecture

Engineering and regulatory requirements must ultimately be traceable to concrete product entities, such as subsystems, components, or interfaces. Figure 2.3 shows how these requirements propagate from abstract stakeholder input to specific technical domains.

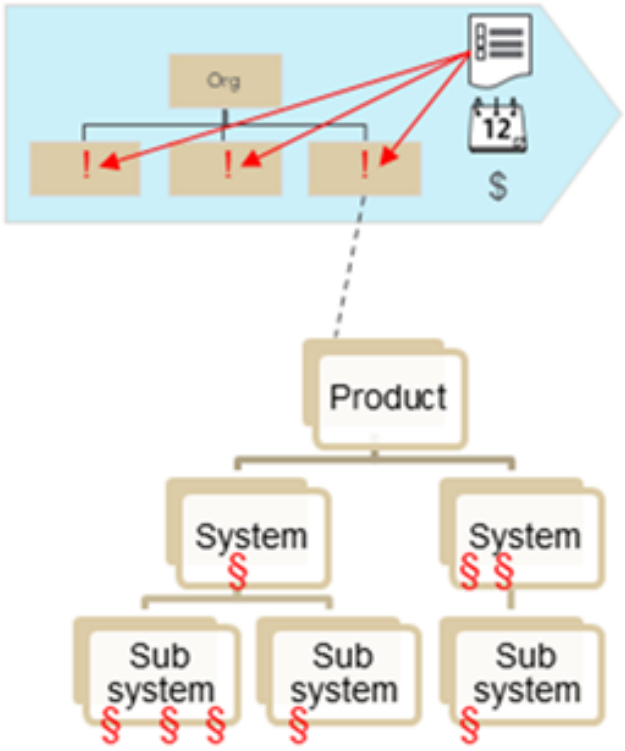

Figure 2.3: Hierarchical mapping of requirements from organisational input to product architecture. The figure illustrates how high-level stakeholder and regulatory inputs are refined and decomposed into technical requirements.

This hierarchical traceability allows:

- Verification to be scoped precisely to affected components.
- Reuse of compliance evidence across product variants.

- Alignment between stakeholder intent and engineering execution.

This decomposition is essential for structured audits, as required by standards such as ISO 26262, and to achieve the presumption of conformity with harmonised EN standards [19, 24].

Focus on Early Compliance Support

As compliance expectations grow, the burden on project teams begins earlier in the lifecycle. Studies have shown that identifying the risks of the requirement during conceptual phases can reduce the total cost of the project and avoid late-stage redesigns [32]. This thesis targets the early gates, especially **FeG** and **CG** by automating the mapping of legal clauses to the TD. The goal is to identify which requirements are *applicable* and *binding* based on product usage context, thereby helping engineers scope compliance efforts as early as possible.

2.4 SystemWeaver

SystemWeaver is Volvo Penta's model-based information backbone to manage product development artefacts [33]. It supports traceability, structured authoring, and collaborative configuration control throughout the development lifecycle. All project data, requirements, system components, tests, and compliance judgments, are represented as typed nodes and links in a version-controlled graph database. Figure 2.4 illustrates the SystemWeaver data model.

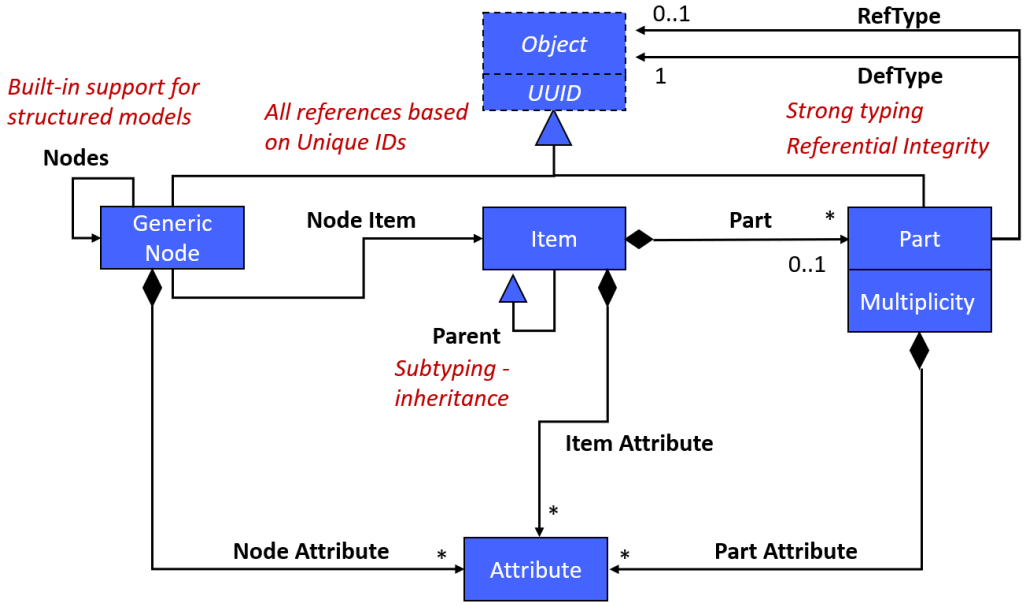

Figure 2.4: Core elements of the SystemWeaver information model [34].

Each artefact, such as a requirement, interface, clause, or test case, is stored as an `item` with a globally unique identifier. These items may contain typed `parts`, and

both items and parts can carry `attributes` that encode relevant metadata. For example, the hierarchical requirements structure shown in Figure 2.3 would be represented as a set of linked `items` in this model. This typed graph structure enables deeply nested artefacts to be traced, versioned, and reused across programmes, while preserving full referential integrity.

Clause Decomposition and Compliance Labelling

Regulatory texts and supporting harmonised standards such as **EN 1175** are manually entered into SystemWeaver for analysis and traceability. Rather than inserting an entire document as a monolith, clauses are broken down section by section or chapter by chapter. Each section is created as a typed `item` with a corresponding identifier and source reference.

Compliance engineers then attach a linked `Conclusion` part to each clause or group of clauses, recording a structured compliance judgment. For each `Conclusion`, there will be a statement indicating whether the clause applies. This statement falls into one of the following three categories:

- **Applicable** — the clause imposes a mandatory requirement that applies to the current product configuration.
- **Not Applicable** — the clause is normative but not relevant for the given system, technology, or scope.
- **No Requirement** — the clause is descriptive, explanatory, or editorial in nature and does not entail implementation obligations.

In addition to categorical labels, most conclusions contain rich explanatory text that justifies the judgment. These entries often provide context, engineering assumptions, or cross references to other standards. For example, the clause in EN 1175:2020 Section 4.3.1 specifies:

> *"Electrical systems of trucks powered by lead-acid batteries shall be designed so that all functions operate in the voltage range from 70% up to 120% of the nominal battery voltage."*

A corresponding conclusion in SystemWeaver might read:

> *"70%–120% of nominal voltage 24V is 16.8V–28.8V. The operating voltage of the system shall be 16V–32V according to ISO 16750-2:2023 section 4.2.2, Table 4, Code F."*

In this instance, it is shown how compliance engineers not only categorize a clause but also analyse it in accordance with internal engineering standards and practices. This commentary aids in cross-functional insight and subsequent validation. As shown in Figure 2.5, each clause is structurally linked to its subparts and annotated with a compliance conclusion. This modelling pattern allows requirements to be

reused, traced, and validated across systems and variants.

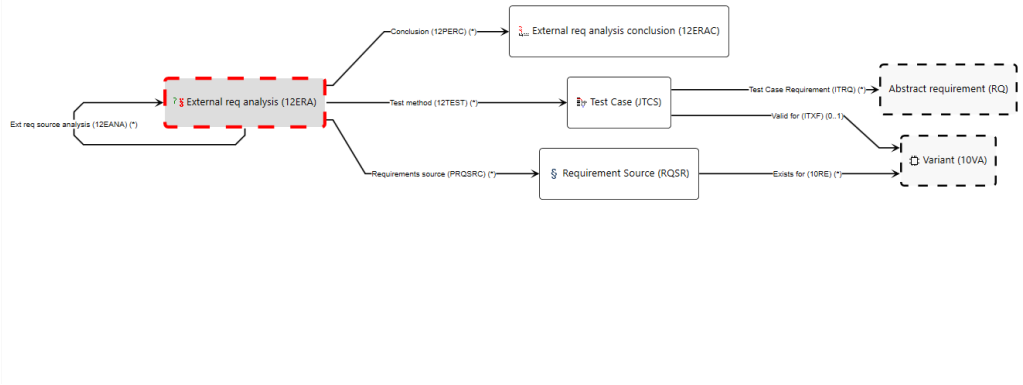

Figure 2.5: Traceability structure for EN 1175:2020 clause 4.3.1 in SystemWeaver. The figure illustrates how the original clause is decomposed and linked to a structured compliance conclusion, relevant test cases, and requirement sources.

2.4.1 Relevance to Compliance Automation

The structured data model of SystemWeaver ensures that all compliance artefacts, requirements, system definitions, and verification plans, are interlinked. This makes it possible to reason about clause relevance in context and to automate applicability predictions. In this thesis, SystemWeaver was used as both a source and destination for structured compliance artefacts. The prototype interacts with SystemWeaver through its public **.NET/C# API**, which enables the extraction of regulatory clauses, requirement structures, and associated metadata. It also includes functionality to automatically upload structured results back into SystemWeaver, maintaining traceability and alignment with existing artefact structures.

2.5 Representation Learning for Textual Compliance

Modern NLP systems convert sentences or paragraphs into fixed-length numerical vectors called *embeddings*, whose geometry captures semantic similarity. If two passages encode conceptually related ideas, their vectors lie close together in the embedding space; unrelated passages map farther apart. This property underpins tasks such as semantic search, clustering, and duplicate-detection in regulatory corpora.

From tokens to vectors

Early embedding schemes (e. g. word2vec [35] or GloVe [36]) assign one vector per word. Contemporary approaches instead learn contextual embeddings, a single encoder processes the entire sentence and produces token vectors that already reflect word order, syntax, and domain terminology. Transformer encoders achieve this with self-attention layers, which let every token weigh its relationship to every other token in the sequence [37]. After the final layer we obtain a matrix of token representations; pooling—mean, max, or a special `[CLS]` token—reduces that matrix to one d-dimensional sentence vector (e. g. 768 d for BERT [38] or 1024 d for Sentence-BERT [39]).

Pre-trained embedding models

A *pre-trained embedding model* is an encoder that has already been optimised on a large generic corpus (Wikipedia, Common Crawl, patents, legal text, etc.) before being applied to a company-specific problem. In this stage the network learns to place semantically related sentences close together in the vector space without seeing any project-specific labels. Examples include models such as BERT [38], the Universal Sentence Encoder [40], and multilingual XLM-R style encoders [41]. Engineers can reuse these models off the shelf by passing each clause through the encoder to obtain a fixed-length vector representation. This enables immediate application of techniques such as similarity search or clustering, without the need to train embeddings from scratch. The same encoder can later be fine-tuned in a smaller domain-labelled data set (e., g. Legal-BERT [42]) to capture the in-house terminology with greater precision.

Similarity and retrieval

Given two embeddings $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, their **cosine similarity**

$$\cos(\mathbf{u}, \mathbf{v}) \;=\; \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \, \|\mathbf{v}\|}$$

provides a bounded ($[-1, 1]$) measure of semantic relatedness that is robust to sentence length and scaling. By embedding every clause of a standard and every paragraph of a product description, one can retrieve *candidate matches* simply by ranking the cosine scores.

Scaling

Practical compliance archives may contain tens of thousands of sections. Exhaustively comparing each new query against all stored vectors is costly. Libraries for *approximate nearest-neighbour* search, such as Facebook AI Similarity Search (FAISS) [43], accelerate retrieval by partitioning the vector space and pruning unlikely matches while maintaining near-perfect recall. The result is millisecond-level response times even on commodity hardware.

Why embeddings help compliance work

- **Terminology robustness**. Embeddings reveal similarity even when two clauses use different wording (e. g. “battery isolation” vs. “galvanic separation”).
- **Language-agnostic analysis**. Multilingual encoders enable alignment between English regulations and Swedish internal documentation without separate dictionaries.
- **Quantitative triage**. Similarity scores give an objective first pass, helping engineers focus manual review on the most promising (or riskiest) clauses.

These properties make representation learning a natural first stage in an automated compliance pipeline: it rapidly narrows thousands of legal sections down to a small, relevant subset that merits deeper reasoning.

2.6 Large Language Models and Prompt Engineering

Large Language Models (LLMs) such as ChatGPT[44] are machine learning models trained on large-scale corpus so that they can understand and generate human language. They process user input and outputs the most likely answer based on a large amount of training data. The output of LLMs is influenced by many factors, such as training data, model parameters, and user prompt. Prompt engineering focuses on organizing and designing the input prompts of LLMs to obtain more accurate model outputs. Different prompt inputs will produce significantly different outputs[45]. Taking the right prompt method can improve model performance, improve accuracy, and provide a basis for the interpretability of LLMs. The following are several commonly used and effective prompt engineering methods.

- **Clear instructions prompting.** Due to the randomness of the output of large language models,vague prompts often lead to broad and unfocused responses. Providing clear instructions and detailed problem descriptions helps the model generate more accurate and relevant answers[45].
- **Role definition prompting.** Defining a role means assigning the model a specific identity to guide its responses. For example: "You are a compliance expert." This sets a clear context and helps ensure the model’s outputs are consistent and aligned with the intended perspective.
- **Zero-shot, one-shot or few-shot prompting.** One-shot or few-shot prompting refers to giving the model one or several reference examples to guide the model output. The model learns the logic from these examples and responds accordingly. Zero-shot prompting, on the other hand, only describes the problem without any examples. However, one-shot or few-shot prompting does not always improve performance. Zero-shot can perform better than One-shot or

few-shot in certain scenarios[46].

- **Chain-of-thought (CoT) prompting.** For some complex problems, Chain-of-thought can significantly improve output accuracy. It breaks the problem-solving process into several steps and guides the model to solve the problem step by step[47]. This method also helps improve interpretability of LLMs, as it shows the reasoning process behind the model's final answer.

On the other hand, when using the API of the LLMs, the output of the model can be controlled by certain hyperparameters. For example, the parameter temperature can control the balance between the randomness and certainty of the output[48]. The lower the temperature, the more deterministic the answer will be.

2.7 Key Terminology: System Qualities in Compliance Automation

To support clarity in evaluating and discussing the proposed compliance system, this section defines several core quality attributes that recur throughout the thesis. These include attributes commonly associated with the usability and trustworthiness of Artificial Intelligence (AI) systems: *traceability*, *usability*, *transparency*, *accuracy*, *interpretability*, and *explainability*.

Traceability

Traceability refers to the ability to track the origin, rationale, and application of a requirement or decision throughout the product development and compliance lifecycle [49]. In this thesis, traceability is implemented by linking each regulatory clause to specific technical artefacts in SystemWeaver, enabling auditors and engineers to follow the decision path from raw regulation to product specification.

Usability

Usability denotes how easily and effectively end users, such as compliance engineers, can interact with the system and derive value from it [50]. This includes factors such as interface design, interaction flow, cognitive effort, and the clarity of system outputs. Usability is enhanced through human-in-the-loop mechanisms, editable outputs, and intuitive display of explanations, aligning with standard user-centred design principles.

Transparency

Transparency is the degree to which the internal processes and decision logic of the system are open to inspection and understanding by human users. Transparent systems expose intermediate steps and rationales to support user trust and validation [51]. In this context, transparency is achieved through semantic similarity

scores, keyword matching, clause applicability labels, and LLM-generated explanations, all of which are visible and reviewable.

Accuracy

Accuracy refers to the correctness of the system's outputs, particularly the classification of regulatory clauses as applicable or not applicable to a given product configuration. It is quantitatively assessed using standard metrics such as precision, recall, and F1-score, based on a ground-truth validation dataset. High accuracy is essential for system reliability and user confidence.

Interpretability

Interpretability is the ease with which a human can comprehend the internal logic, structure, or feature contributions of the system [52]. It concerns how well users can predict or reason about the model's decisions. In this thesis, interpretability is supported by using domain-specific embeddings, keyword-based matching, and structured output schemas that are aligned with engineering reasoning.

Explainability

Explainability is the system's ability to provide human-readable justifications for its decisions or outputs [53, 54]. While interpretability focuses on internal mechanisms, explainability emphasizes the post hoc communication of the reasoning behind specific results. In this thesis, explainability is delivered through natural language outputs generated by large language models using chain-of-thought prompting and embedded rationales.

3

Related Work

This chapter reviews previous research relevant to automating regulatory compliance analysis using NLP, embedding models, and LLMs. The literature is organised into five thematic categories: (1) Traditional rule-based and machine learning approaches, (2) Semantic search and similarity techniques, (3) Embedding-based models for legal texts, (4) Large language models for compliance support, and (5) Hybrid human-in-the-loop systems.

3.1 Traditional Rule-Based and Machine Learning Approaches

Early approaches to regulatory compliance classification relied heavily on rule-based NLP and classical supervised learning techniques. These methods used predefined linguistic patterns, heuristics, or manually curated ontologies. For example, Kiyavitskaya et al. [55] developed Gaius T, a semantic annotation framework to extract stakeholder obligations from complex regulations such as Health Insurance Portability and Accountability Act of 1996 (HIPAA) and the Italian Stanca Act. Their method significantly reduced human annotation effort by formalising heuristics for obligation extraction.

Similarly, Breaux and Anton [56] combined ontologies and deontic logic for automatic compliance check in the utility industry, capturing domain-specific terminology and semantic structures. However, these rule-based systems generally required extensive expert input and were limited by their scalability and adaptability to new regulatory texts.

In parallel, supervised machine learning techniques such as Support Vector Machines (SVM), decision trees, and Naïve Bayes classifiers were extensively explored for compliance-related classification tasks. Zhang et al. [57] demonstrated the feasibility of SVM classifiers to evaluate website privacy policies against the PIPEDA regulations of Canada, achieving up to 88% precision. Their approach used term frequency features and Principal Component Analysis (PCA) to reduce dimensionality, establishing early evidence of Machine learning (ML) efficacy in regulatory compliance contexts.

More recently, Fahad ul Hassan et al. [58] studied NLP preprocessing (lemmatisation, Part-of-speech (POS) tagging) combined with traditional ML models (e.g., SVM) to classify construction contract documents. Their methodology effectively distinguished between functional and non-functional requirements, achieving a recall of 95%. Despite high accuracy, their approach highlighted significant limitations due to its heavy reliance on large annotated datasets, limited generalisability across domains, and the considerable feature engineering effort required.

In general, traditional approaches established important foundations but often struggled with deeper semantic nuances and the scalability necessary for modern compliance analysis tasks.

3.2 Semantic Search and Similarity Techniques

To address scalability issues and better capture nuanced regulatory contexts, researchers increasingly adopted semantic search and similarity-based methods. These techniques represent regulatory clauses in semantic vector spaces, enabling context-sensitive document retrieval and classification.

Salama and El-Gohary [59] leveraged domain ontologies for semantic classification of construction-related regulatory texts, significantly improving efficiency and precision in compliance check. Similarly, Zhou and El-Gohary [60] introduced multi-label semantic classification to environmental regulations, accommodating complex cases where single clauses address multiple compliance categories.

Sleimi et al. [61] presented a semantic NLP framework tailored specifically to Luxembourg's legislation, achieving high precision in metadata extraction and systematic annotation. However, the heavy reliance of their system on jurisdiction specific annotated data limited its broader applicability, highlighting the difficulty of generalising semantic solutions across diverse legal systems.

Amaral et al. [62] further advanced semantic compliance check by embedding General Data Protection Regulation (GDPR) provisions and data processing agreements (DPA) in vector spaces, employing cosine similarity for automated compliance assessment. They reported a successful identification of missing regulatory provisions, showcasing embedding-based semantic matching effectiveness.

Additionally, Chalkidis et al. [63] proposed a two-step information retrieval system for EU and UK legal compliance, first employing traditional keyword-based retrieval (BM25), then re-ranking results with neural semantic embeddings (Legal-BERT). Their method notably improved relevance over purely lexical-based retrieval, emphasising the critical role of domain-specific embeddings in semantic compliance analysis.

In addition, Zhou et al. [64] explored integrating NLP and context-free grammars to formalise and interpret regulatory texts. Their approach provided structured semantic extraction, but struggled with the ambiguities and complexity inherent

in unstructured regulatory language. This underscores the necessity of combining semantic methods with more robust machine learning techniques to develop scalable compliance solutions.

3.3 Embedding-Based Models for Legal Text

Embedding-based NLP methods significantly improved semantic understanding in regulatory contexts, overcoming the limitations of traditional methods. Pre-trained embeddings such as Word2Vec, GloVe, and FastText enabled richer semantic representations, capturing subtle conceptual relationships in legal language.

Wen et al. [65] demonstrated the advantage of Word2Vec embeddings over Term Frequency–Inverse Document Frequency (TF–IDF) in distinguishing compliant from non-compliant corporate policy documents, achieving a notable F1 score of 0.84. These static embeddings, while efficient, still faced challenges in capturing varying contextual meanings across different legal scenarios.

Contextual embeddings, particularly transformer-based models like BERT and domain-specific Legal-BERT, further improved semantic comprehension. Chalkidis et al. [42] established the effectiveness of Legal-BERT, significantly exceeding the generic BERT embeddings in EU legal tasks.

Furthermore, Amaral et al.[13] successfully employed SBERT (Sentence-BERT) embeddings for GDPR compliance verification, achieving impressive precision, recall, and F1 scores around 81%. Similarly, Zhong et al.[66] summarised multiple embedding-based applications, reinforcing the value of fine-tuning embeddings on annotated legal datasets to optimise performance.

Finally, Moon et al.[67] fine-tuned BERT on 2 807 construction-risk clauses, reporting 88.9% accuracy and an F1 of 0.93, while Braun & Matthes [68] attained approximately 83% accuracy with F1 of 88% on multilingual terms-and-conditions clauses. These results set a performance band of 79–89 % accuracy for clause-level tasks against which we later compare our own accuracy.

3.4 Large Language Models for Structured Compliance Support

The advent of LLMs such as GPT-3, GPT-4, Claude, and LLaMA represented a transformative advance in compliance analysis capabilities. LLMs expanded beyond classification and retrieval tasks to include natural language generation, nuanced interpretation, and structured reasoning.

Balasubramanian [69] evaluated GPT-3.5's capabilities in answering complex regulatory questions, demonstrating high accuracy and human-level acceptability. Hassani [70] applied GPT-4 for GDPR compliance verification, achieving 81% accuracy

and substantial efficiency improvements (0.7 seconds per paragraph processed).

Wu et al.[71] introduced Reg-LLaMA, a variant fine-tuned specifically to financial regulatory data, significantly outperforming base LLaMA models in specialised legal summarization tasks. Similarly, Cao and Feinstein [72] evaluated the effectiveness of GPT-4 in interpreting Basel III banking regulations, emphasising prompt engineering and structured validation methods to improve reliability and reduce hallucinations.

Despite significant capabilities, LLM deployment in compliance contexts remains cautious due to ongoing issues such as hallucinations, inconsistencies, and lack of grounding in external references. Researchers advocate using prompt engineering, structured reasoning (chain-of-thought) and human validation to mitigate these limitations.

3.5 Hybrid Human-in-the-Loop Pipelines

Due to inherent complexities and high stakes in compliance analysis, fully automated systems remain impractical for many industrial applications. Consequently, hybrid human-in-the-loop (HITL) systems have emerged as best practices, combining AI-driven initial analyses with expert human oversight for validation and interpretation.

Wen et al.[65] implemented a HITL framework for corporate compliance, using semantic thresholds for AI-generated matches reviewed and refined iteratively by legal analysts. Similarly, Hassani[70] combined keyword-based methods with LLM predictions, reconciling discrepancies through human curation.

Industry implementations such as RadarFirst [73] emphasize AI's supportive role, recommending human experts ultimately resolve flagged compliance risks. Active learning paradigms further enhance efficiency, querying human experts only in ambiguous or novel regulatory cases to optimize annotation workload while maintaining high reliability and accountability.

This approach is increasingly recommended by regulators and standard-setting bodies, highlighting that AI systems should augment, but not replace, human judgment in regulatory compliance contexts.

4

Methods

This chapter will thoroughly describe the implementation of the methodological framework. Our methodology follows the pipeline layout below and is divided into three main parts. The first part includes data preprocessing, embedding generation, and semantic matching. This corresponds to **RQ1**, which focusses on linking regulatory requirements with relevant product and engineering information. The second part involves generating conclusions combined with Human-In-The-Loop review and exporting the results to SystemWeaver, which corresponds to the integration with existing workflows of **RQ2**, and it also emphasises traceability, usability, and transparency. The third part is the part for verifying the results, which corresponds to **RQ3**. In this part, we combine quantitative and qualitative verification methods, including the model performance matrix and expert evaluation. The full pipeline can be seen in figure 4.1.

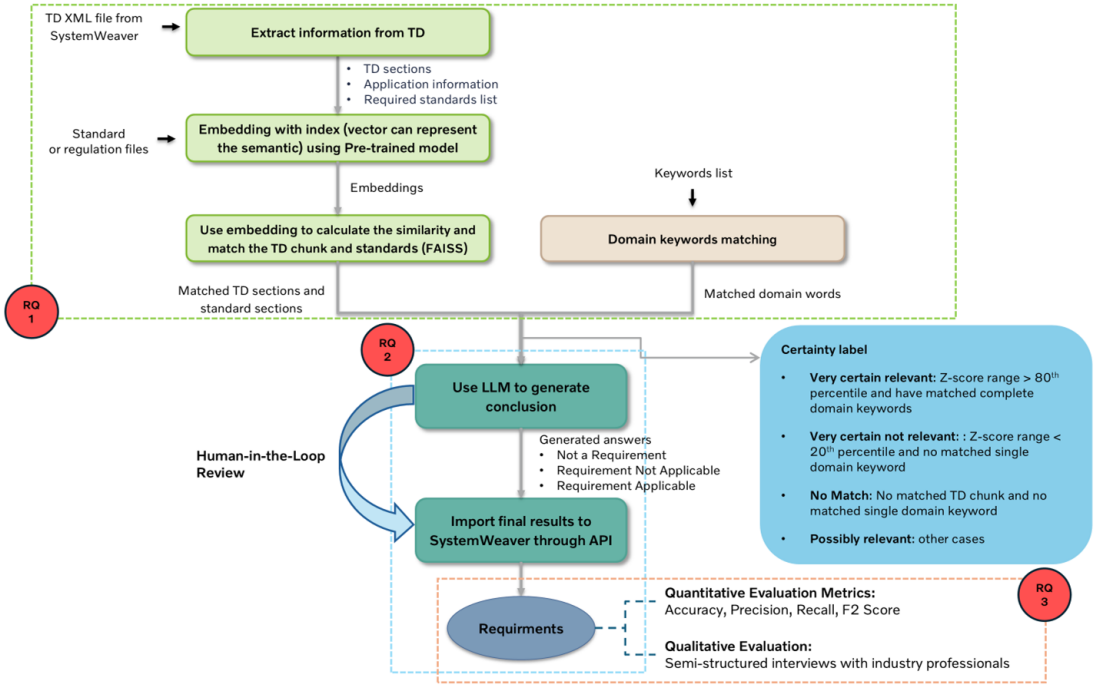

Figure 4.1: NLP-based compliance requirements generation model pipeline.

In addition, we also provide clarity on the technological choices, justifications for the selected methods, and a clear description of the evaluation strategy used to validate the effectiveness and precision of our proposed solution. All implementation was developed entirely in Python, version 3.11.11.

4.1 Research Methodology

This thesis adopts an Action Research methodology, following the principles outlined by Runeson et al.[74]. Action Research is particularly well-suited for applied software engineering research in industrial settings, allowing close interaction between researcher and practitioner in iterative cycles of planning, action, observation, and reflection. This is consistent with the classification framework of Stol and Fitzgerald[75], who distinguish Action Research as a highly collaborative, context-driven approach well aligned with real-world problem solving.

The research process followed these main steps:

- **Diagnosis:** Identify bottlenecks in manual compliance interpretation at Volvo Penta.
- **Planning:** Design an automation pipeline using NLP, semantic similarity, and LLM techniques.
- **Action:** Develop and deploy a prototype system integrated with SystemWeaver.
- **Evaluation:** Use accuracy metrics, case studies, and expert feedback to validate the system.
- **Reflection:** Assess impact, extract lessons learnt, and identify directions for industrial adoption.

This approach ensures that the solution is both empirically grounded and practically useful.

4.2 Data Sources

The study draws on four complementary data sources that mirror the artefacts Volvo Penta engineers handle during a typical compliance assessment workflow: (1) formal regulatory documents, (2) internal product documentation, (3) an expert-labelled ground truth file, and (4) a domain-specific keyword lexicon. Table 4.1 and Table 4.2 list descriptive statistics.

In practical industrial settings, the set of regulatory standards applicable to a project is not static or universal. For each new product or platform, a dedicated team, typically from Regulatory Affairs or Compliance, selects a subset of directives, harmonised standards, and internal specifications deemed relevant to the project con-

text. This selection is guided by factors such as the product application domain (e.g., marine, off-road), target markets, and technical characteristics. The selected set is formally referenced in the Target Description (TD), which serves as the definitive scope for early compliance activities.

Although harmonised standards such as EN 1175 are derived to support compliance with overarching EU directives (e.g., 2014/30/EU for EMC), they do not replace the directives themselves. Directives remain the binding legal instruments and often include general safety principles and cross-domain obligations not fully addressed in any single standard. In practice, engineers consult both the directives and their associated standards to ensure complete coverage and legal defensibility. For this reason, this study includes both types of documents in its analysis.

The specific regulatory documents included in this study represent a subset of the standards formally cited in the TD of the anonymised electric propulsion platform project introduced in Section 4.2.2. This selection was limited by two key factors. First, only a portion of the referenced documents were available in a structured digital format suitable for clause-level processing. Second, among those, EN 1175 was the only standard that had already been fully analysed and annotated with compliance conclusions by domain experts. As such, it served as a practical starting point for system development and validation. Although this limits coverage relative to the complete TD, it reflects realistic constraints in industrial data access and still allows representative evaluation of the proposed approach across multiple regulatory domains.

Consequently, this thesis assumes that the relevant standards for a given project have already been identified and documented in the TD. The automation pipeline is then applied to assess clause-level applicability within that predefined regulatory scope.

4.2.1 Regulatory Documents

- **SS-EN 1175:2020 Standard:** The official version of the EN 1175:2020 standard, issued by the Swedish Institute of Standards (SIS), was obtained as a PDF document licenced. This 103-page harmonised standard supports compliance with applicable legal requirements for electrical and electronic safety requirements applied to industrial trucks in this project. Provided in a pre-segmented Excel file containing 256 normative clauses. Each clause includes an external ID, title, and comprehensive textual description, facilitating structured semantic analyses. Retained from SystemWeaver internally.

- **Horizontal EU Directives:** Three directives were included: EMC Directive 2014/30/EU, Low-Voltage Directive 2006/95/EC, and Battery Directive 2006/66/EC. These were parsed from HTML editions of the Official Journal, contributing 259 regulatory clauses.

4.2.2 Internal Project Documentation

The internal documentation refers to an anonymised electric propulsion platform used in a real-world compliance context.

- **Target Description (TD):** Provided as an XML export containing 239 sections, including technical and functional content. Retained from SystemWeaver internally. The TD also includes a dedicated *Standards* section that formally cites regulatory frameworks and harmonised standards relevant to this project. A complete list is provided in Appendix A.

- **Application Summary:** A section embedded within the TD identifying 34 binary features (e.g. *"Marine commercial"*, *"Road-legal"*) using Unicode checkbox markers to indicate presence or absence.

4.2.3 Ground-Truth Dataset

An expert-validated ground truth dataset derived from a completed compliance audit with EN 1175 is used for model validation and evaluation. The dataset is based on the internal documentation described in Section 4.2.2. Labelling was not performed directly by experts in the domain. Instead, it was derived from the conclusions written by systems engineers with extensive experience in standards compliance and system architecture. Each clause was associated with a conclusion based on its relevance to a specific project. For example, if a conclusion stated that a clause was relevant for Penta, it was labelled applicable (1); otherwise, it was marked as not applicable (0).

This dataset contains clause–project classifications into three categories: *Applicable*, *Not Applicable*, and *No Requirement*. These labels were derived from the compliance conclusions associated with each clause, as described earlier. A total of 256 clause–project pairs are included, comprising 138 positive examples (*Applicable*, 54%), 89 negative examples (*Not Applicable*, 35%), and 29 entries labelled as *No Requirement* (11%), which denote descriptive or editorial clauses that do not impose implementation obligations.

4.2.4 Domain Keyword Lexicon

The process of semantic matching relies on a carefully compiled lexicon consisting of 226 distinctive domain-specific terms. This lexicon was crafted through three collaborative workshops with specialists in electrical systems, battery technologies, and electromagnetic compatibility (EMC). It is organised into 14 separate technical categories and covers hardware, software, performance, safety, and drive-related elements of energy storage systems. For purposes of transparency, traceability, and ease of maintenance, the lexicon is maintained in plain text format within the analytical workflow. The entire lexicon can be found in Appendix B.

4.3 Preprocessing

To ensure that diverse input documents could be consistently analysed, a structured preprocessing pipeline was applied to all data sources. The objective was to transform each input into a semantically coherent, metadata-rich corpus of text sections, suitable for embedding, semantic search, and LLM-based analysis.

4.3.1 Document-Specific Pre-processing Workflows

- **Standard (EN 1175).** The Excel version of the EN 1175 standard served as the primary input for regulatory requirements. Each row or section contained a clause ID, title, and full description. Using `pandas`, empty fields were completed using fallbacks from adjacent columns, and a flat table of 256 normative sections was produced.

- **Target Description (TD).** The technical documentation was exported as a structured XML file from SystemWeaver. All nodes were extracted along with their metadata (ID, section text, and ancestor links). Irrelevant categories such as placeholders and template structures were filtered on the basis of metadata fields.

- **EU Directives.** Legal directives were downloaded from the Official Journal in HTML format. The main content subtree was extracted using `BeautifulSoup`, retaining all legal text while removing scripts, styles, and boilerplate elements such as headers and footers.

4.3.2 Section Structuring and Normalisation

Each document was segmented into meaningful text sections. For documents with formal structure (e.g., clause numbers or XML items), one section per clause or node was defined. Key operations included:

- Removal of trailing punctuation, redundant whitespace, and non-informative formatting.

- Normalisation of checkboxes: visual glyphs such as [X] and [ ] were replaced by `[TRUE]` and `[FALSE]`, respectively.

- Expansion of standard abbreviations (e.g., `Art.` → `Article`).

4.3.3 Structured Representation

All preprocessed data was stored in structured tabular formats to ensure consistency across downstream processing. Two primary schemas were used:

Standards and Requirements These were stored with the following fields:

```
section_id         -- Unique identifier for the section
section_title      -- Title or heading of the section
standard_text      -- Full textual content of the standard
source_standard    -- Source of the requirement (e.g., EN 1175, Directive)
is_binding         -- Boolean flag indicating legal binding status
```

Target Descriptions These were organised using:

```
item_id               -- Unique identifier for the item
sid                   -- Section or subsection identifier
ancestor              -- Parent or hierarchical ancestor
product_section_title -- Title of the relevant TD section
product_section_text  -- Full text content of the description
is_placeholder        -- Boolean flag indicating if content is a placeholder
```

4.3.4 Binding Clause Identification

Binding status was determined heuristically based on structured patterns in the source material. The following rules were applied across the three document types:

1. **Section Numbering:** For structured standards such as EN 1175 and ISO 14990, sections appearing in introductory chapters (1-3) were marked as non-binding.

2. **Title and Prefix Heuristics:** Titles containing terms like *Scope*, *Foreword*, or *Annex*, or HTML sections with known prefixes (e.g., `rct_`, `pbl_`) were marked as non-binding.

3. **Template Metadata:** In the TD XML, elements with known template names (e.g., document headers or boilerplate sections) or placeholder-like descriptions were excluded from downstream processing.

Table 4.1: Structured regulatory and internal sources before and after preprocessing. "Total sections" refer to entries structured according to Section 4.3.3. "Marked valid" includes legally binding clauses and non-boilerplate.

Source	Total sections	Marked valid
EN 1175 (Excel)	256	227
Low Voltage Directive (HTML)	76	32
EMC Directive (HTML)	115	47
Battery Directive (HTML)	68	30
Target Description (XML)	239	173

Table 4.2: Supporting resources for compliance analysis.

Resource	Count
Application Summary Features (Checkbox)	34
Expert Ground-Truth Clause–Project Pairs	256
Domain Keyword List	226 unique terms

4.4 Model Construction

Following preprocessing, the pipeline operates on two aligned corpora:

- $\mathcal{P}$ — $N_p = 173$ product sections of the internal documentation.
- $\mathcal{S}$ — $N_s = 515$ regulatory clauses originating from Excel and HTML.

The objective is to map each clause $s \in \mathcal{S}$ to one of the three compliance outcomes: *Not a Requirement*, *Requirement Not Applicable*, or *Requirement Applicable*, accompanied by a certainty label. Notably, sections labelled as "non-binding" remain part of the processing pipeline. This supports the goal of full coverage and traceability.

4.4.1 Embedding and Semantic Search

All text sections were transformed into vector representations using embedding models. For standards and regulatory clauses $s \in \mathcal{S}$, only the title of the section and the body text were used to form the input, concatenated as `text_for_embedding = section_title + " " + section_text`. The same approach was applied to the product sections $p \in \mathcal{P}$, using their respective titles and texts. This consistent structure ensured a uniform semantic representation across both corpora while excluding metadata fields. This embedding input format aligns with the structured fields defined in Section 4.3.3.

To explore the impact of different embedding model architectures and intended use cases, we selected two retrieval-optimized models: Legal-BERT, trained specifically

for legal and regulatory language, and BGE-m3, a contrastively trained encoder known for strong semantic retrieval performance. Additionally, we included two general-purpose embedding models from major AI companies: text-embedding-ada-002 from OpenAI and databricks-gte-large-en from Databricks. These embeddings were used to identify semantic relationships between regulatory clauses and internal content. The details of these four embedding models are shown below:

- **Legal Bert:** A domain-specific BERT model fine-tuned for legal texts and applications[42].
- **BGE-m3:** A third-generation model from the BGE family, optimised for multilingual and dense retrieval tasks[76].
- **text-embedding-ada-002 (OpenAI): (default)** A production grade embedding model accessed through Azure OpenAI, which produces 1,536-dimensional vectors[77].
- **databricks-gte-large-en (Databricks):** A transformer-based model built for general-purpose semantic search and retrieval tasks[78].

All vectors were L2-normalized and stored in separate FAISS `IndexFlatIP` indices, one per source document (e.g. EN 1175, EMC directive). This design allows cosine similarity to be calculated via the inner product, leveraging L2-normalization. FAISS was selected for its scalability and performance, enabling efficient top-k retrieval even when operating across multiple clause sets independently. By maintaining a distinct index for each standard or directive, we ensure that clause–product matching is not biased by size or density differences across sources. This separation improves recall balance and allows clause relevance to be judged within the contextual boundaries of each source document.

4.4.2 Domain Keyword Matching

Due to the domain-specific nature of the words on the list, it is important to accurately detect their presence in the regulatory clauses. Instead of calculating similarity, we use keyword matching to perform this search.

The keyword list contains three types of lexical items:

- **Technical word.** For example, *"torque", "radial", etc.*
- **Domain-specific phrases.** For example, *"Power supply", "short circuit", etc.*
- **Descriptive phrases.** For example, *"extra low voltage", etc.*

For the first two types, words and phrases may appear in various forms in sentences, such as singular or plural forms, or as gerunds. In addition, for phrases, we are not limited to exact matches; individual words within a phrase can also carry important meaning, for example *"circuit"* in *"short circuit"*. For the third type, there may be

different ways to express the same concept. For example, *"extra low voltage"* may also appear as *"the voltage is extremely low"*. Based on these considerations, we adopt a two-step approach: first match individual words, and then combine them into phrases based on context.

First, we split all the regulatory texts and keyword lists that need to be matched into single words. In order to improve the matching efficiency, we remove stop words. After that, all words are lemmatised, while proper noun abbreviations (e.g., *"EMC"*) are preserved in their original form. After word matching, for each sentence of the regulatory clause, we compare the matched words with the keyword list. If the matched words form a phrase that exists in the keyword list, the complete phrase is extracted as the matching result. Finally, two matching information are generated for each regulatory clause: *"Matched_Domain_Words_single"* shows the single word matching result. *"Matched_Domain_Words_complete"* field shows the complete result of the matching of phrases.

Using the 226 unique term lexicon, the procedure yields 297 single-token hits and 178 full-phrase hits across the EN 1175 standard and three EU directives. A document-level breakdown (one standard + three directives) is provided in Appendix C.1, Table C.1.

4.4.3 Semantic Retrieval of Regulatory Matches

Once all product and standard sections are embedded, each product section $p \in \mathcal{P}$ is matched to a corresponding regulatory source $\mathcal{S}_i$ via semantic similarity. The embedding input for product sections mirrors that of the standards, as described in Section 4.4.1.

Each product embedding $\hat{\mathbf{p}}$ is matched independently against each index, ensuring that clause relevance is evaluated within the context of its own source. Because all vectors are L2-normalized, the inner product computed by FAISS corresponds directly to cosine similarity. This approach avoids bias introduced by corpus size or density and provides more balanced recall across heterogeneous regulatory texts.

We define two key retrieval parameters: $k = 10$ (top-k results) and a similarity threshold $\sigma \geq 0.5$. These values were selected after empirical testing with multiple configurations ($k = 5$, 10, 20 and σ ranging from 0.3 to 0.7), with the aim of balancing retrieval quality and computational efficiency. The value $k = 10$ was found to offer sufficient recall while keeping the number of candidate clauses tractable for downstream processing. The threshold $\sigma \geq 0.5$ helps eliminate surface-level matches lacking real semantic alignment. These settings ensure the method remains scalable and responsive in industrial contexts. Furthermore, because the end goal is to input these into an LLM these parameters were deliberately chosen to be moderately inclusive. The goal at this stage is not to make definitive compliance decisions, but to assemble a broad set of candidates.

For each match above the threshold, associated metadata and domain keyword

matches are collected. Specifically, matched domain words, both individual tokens and complete phrases, are appended to the result. All matches are stored as structured rows in a result data frame, as shown in Figure 4.2. This data frame serves as the foundation for subsequent classification and prompt construction.

Using these parameters with our encoder (`text-embedding-ada-002`) returns 197 candidate clauses for EN 1175 (176 of them binding), 67 for the Low-Voltage Directive (28 binding), 92 for the EMC Directive (34 binding), and 66 for the Battery Directive (31 binding). A document-by-document breakdown for all four encoders evaluated in this study is given in Appendix C.2, Table C.2.

```
{
  "section_id": ...,
  "section_title": ...,
  "standard_text": ...,
  "source_standard":...,
  "is_binding": true | false,
  "matched_domain_Words_single": ...,
  "matched_domain_Words_complete": ...,
  "matched_product_section_text": ...,
  "similarity": ...,
}
```

Figure 4.2: Schema of the result dataframe storing clause–product matches.

4.4.4 Candidate Aggregation and Certainty Labelling

Once the candidate matches are recovered, the goal is to aggregate and classify them according to relevance. The retrieval process from Section 4.4.3 is used to identify the most similar product sections for each regulatory clause s. For each clause, we retain the three most relevant sections, concatenate their texts, and compute the maximum similarity score $\sigma_{\max}(s)$ across the retrieved set.

Although the final analysis is framed around each regulatory clause, the matching process initially iterates over product sections. This design choice is motivated by computational and practical efficiency. The number of product sections is typically smaller and more stable than the potentially large and diverse pool of regulatory clauses spanning multiple standards. Matching from the product side allows early filtering of irrelevant content, accelerates similarity search, and avoids overwhelming the system with low-quality matches. After collecting high-confidence matches from the product perspective, this step effectively flips the perspective of the analysis. Instead of continuing to focus on product sections, we now aggregate results per standard clause. This reorientation is intentional and aligns with industrial practice, compliance assessments are typically structured around verifying regulatory obligations one-by-one, not reverse-mapping from internal documents. By focusing on each standard clause, we can evaluate whether sufficient contextual evidence ex-

ists in the product documentation to support or exclude its applicability. Clauses without strong semantic alignment are retained in the dataset for completeness but are assigned a similarity score of zero.

Each clause is uniquely identified by its combination of `section_id`, `section_title`, and `standard_text`, allowing matches to be reliably grouped and aggregated across sources. The aggregation process groups the clause–product matches by these keys and combines their top 3 matching evidence sections. This ensures that we capture diverse contextual support for each clause while maintaining tractability. After merging with the full regulatory set, the resulting data frame contained 515 total clauses, including those without high-similarity matches. Clauses without semantic matches are explicitly retained in the dataset with a zero similarity score.

To normalise scores across heterogeneous content lengths and domains, a z-score is calculated for each $\sigma_{\max}(s)$ based on the distribution of non-zero similarities. The 20^{th} and 80^{th} percentiles of the z score distribution are used together with the keywords from 4.4.2 to define adaptive certainty labels. The decision logic is summarised in Table 4.3.

Table 4.3: Certainty labelling logic based on similarity z-scores and domain keyword matches.

Z-Score Range	Keyword Indicator	Certainty Label
$= 0$ (no matches)	None	No Match
$< 20^{\text{th}}$ percentile	No domain unigram match	Very certain not relevant
$> 80^{\text{th}}$ percentile	Full-phrase domain match	Very certain relevant
Otherwise	Any	Possibly relevant

A document-level overview of the certainty labels is provided in Appendix C.3, Table C.3. The numbers of clause–section matches that fall in the lower (20 %) and upper (80 %) similarity bands, are summarised in Table C.4 of the same Appendix. This approach ensures that the certainty labels are statistically adjusted to the observed distribution of similarity scores. The result is a data frame with clause metadata, similarity scores, aggregated product evidence, keyword hits, and certainty classifications (see Figure 4.3).

```
{
  "section_id": ...,
  "section_title": ...,
  "standard_text": ...,
  "source_standard":...,
  "is_binding": true | false,
  "matched_Domain_Words_single": ...,
  "matched_Domain_Words_complete": ...,
  "matched_product_section_text": ...,
  "similarity": ...,
  "z_score":...,
  "certainty_label": "Very certain relevant" |
                     "Possibly relevant" |
                     "Very certain not relevant" |
                     "No match"
}
```

Figure 4.3: Schema of the final aggregated dataframe after candidate matching and certainty labelling.

4.5 LLM-Assisted Requirement Classification

This section describes the use of a LLM in the final stage of our pipeline, where each regulatory clause is assessed for applicability against project documentation. Due to the diversity of input information and the complexity of reasoning, one-shot or few-shot prompting is not suitable for this task.

4.5.1 Prompting Strategy

To help the model reach conclusions through a reasoning process aligned with compliance work and considering the need for interpretability in engineering applications, we use a combination of zero-shot and Chain-of-Thought (CoT) prompting. To provide a system-level background, we combine role definition with clear instructional guidance. The general flow of reasoning and prompting is shown in Figure 4.4.

4.5.2 Prompt Construction

The entire prompt construction process is as follows:

1. **Background definition.** First, the model is assigned the role of a legal / regulatory compliance advisor that helps meet the product requirements. Second, the model is informed of the structure of the user input and its main task. At the same time, background information about the project is provided.

2. **Implement Chain-of-Thought (CoT) prompting.** The model is clearly

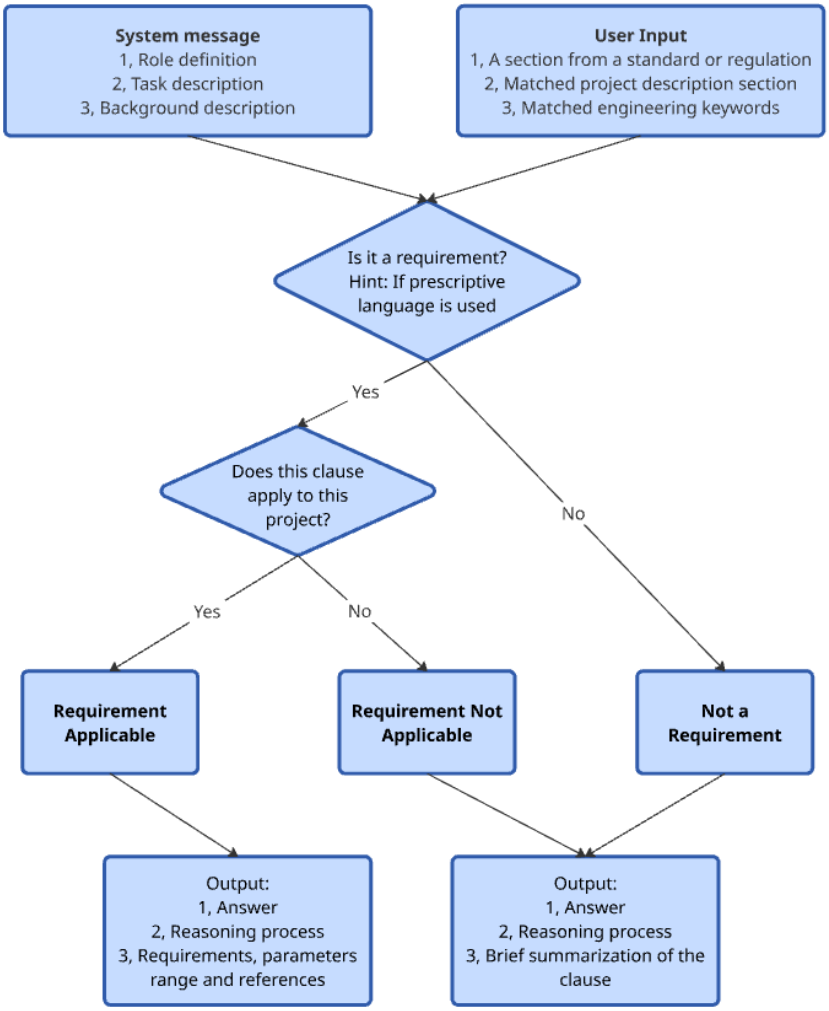

Figure 4.4: Prompting and reasoning flowchart in LLM.

instructed to choose from three possible answers: Not a Requirement; Requirement Not Applicable; Requirement Applicable. The reasoning process for answering questions is as follows: First, the model determines whether this regulation or standard clause constitutes a requirement. A guiding prompt is provided: If the clause uses prescriptive language ('shall,' 'must,' 'requires,' 'obligates'), it is considered as a requirement. Second, if this clause constitutes a requirement, the model needs to determine whether this requirement is applicable to the project. The basis for judgment is the relevance between the clause content and the project description keywords provided by the user. Here, the model is also given background information on the company's business scope and is also required to refer to the model in the judgment process.

3. **Result output.** The result output will consist of three parts. *1) The answer, 2) The reasoning process, 3) The term explanation.* For the third part, the model will generate different answers based on the judgment results of the previous step. If the answer is Requirement Not Applicable or Not a Requirement,

the model will output a summary of this clause. If the answer is Requirement Applicable, in order to improve the implementability of this conclusion in the engineering practice process and help people quickly understand this clause, the model is required to identify the subject of the clause and the specific requirements for this subject. At the same time, the model is also required to list the range of values of specific parameters and all references (if mentioned). Here, the one-shot prompt is used to help the model understand the range of values of a specific parameter.

The full implementation code and system prompt logic are provided in Appendix D.

4.5.3 Execution Settings

During the compliance checking process, the answer to whether a regulatory clause applies to a project should be deterministic. At the same time, since the input prompt, including the matching project description and keywords, are unchanged, the reasoning process for generating this answer should also be consistent. Based on this consideration, the low temperature setting was chosen to enhance the consistency of the response, ensuring that the model produces stable and reproducible outputs, particularly important for regulatory tasks that require high interpretability and reliability.

While setting `temperature = 0` would theoretically maximize determinism, it can lead to overly rigid or repetitive phrasing and does not fully guarantee determinism in practice. Recent research has shown that outputs can still vary slightly even at zero temperature due to floating-point precision issues and hardware-level nondeterminism in LLM backends [79]. Therefore, we used a slightly higher setting of `temperature = 0.2`, which preserves consistency while allowing for clearer and more fluent justifications in clause reasoning.

Each aggregated record sends a message to the LLM with `max_tokens = 800`. Nonbinding clauses skip the LLM and receive "Not a requirement" by default.

4.5.4 Evaluated Language Models

To evaluate the impact of different model architectures and sources on the results, and identify the most effective combination of embedding models and LLMs, we performed a comparative analysis of the validation set using the four embedding models introduced in Section 4.4.1 and four LLMs. Referring the research by Hassani et al.[70], the three widely used and well-regarded models: GPT-4o-mini, GPT-3.5-turbo from OpenAI and Meta-llama-3-70b-instruct from Meta are selected. In addition, to explore the latest advancements of LLMs, we included Claude 3.7 Sonnet from Anthropic, a powerful new LLM released in 2025. The details of the four models are shown below:

- **GPT-4o-mini** (OpenAI): A compact and efficient version of GPT-4o, offering strong performance with faster inference and lower cost, suitable for scalable

compliance automation[80].

- **GPT-3.5-turbo** (OpenAI): A widely used baseline model with strong language understanding capabilities, used here as a point of comparison for more advanced architectures[81].
- **Meta-llama-3-70b-instruct** (Meta): A 70 billion-parameter instruction-tuned model from Meta, evaluated for its robustness in complex reasoning tasks in regulatory language[82].
- **Claude 3.7 Sonnet** (Anthropic): A safety-aligned, instruction-following model with high transparency and strong performance on legal and interpretive tasks[83].

4.5.5 Final Output Schema

The output of the compliance pipeline is a structured data frame that contains all relevant metadata, similarity scores, domain keyword matches, LLM-based classifications, and certainty labels. This data frame serves as the final result passed to downstream users and systems, such as SystemWeaver, for compliance documentation and traceability.

Importantly, the pipeline uses a two-step filtering approach: first, semantic similarity is used to identify potentially relevant matches between the regulatory standard and the product description. However, not all of these matches are actual requirements; some may be contextually related without imposing a concrete obligation. Therefore, the second step applies a large language model to interpret each matched pair and generate a structured conclusion, classifying whether the clause genuinely constitutes a requirement and, if so, in what form. This layered process ensures high recall while reducing false positives, helping downstream users focus only on actionable obligations.

Each row in the output represents one clause–product pairing and includes the following fields:

This structured format supports both traceability and further automation. It allows downstream systems not only to retrieve model predictions but also to review the justification logic (via LLM answers), evaluate matching strength, and audit domain keyword relevance. The inclusion of prediction and ground truth fields also enables robust model evaluation, as discussed in Section 4.7.

4.6 Prototype Integration

This section outlines how the research pipeline described in Sections 4.4 and 4.5 was operationalised into a practical end-to-end compliance tool within Volvo Penta. The implementation is designed to preserve traceability and modularity while offering an accessible front-end for legal and engineering teams.

Table 4.4: Final output schema of the compliance pipeline.

Field Name	Description
section_id	Unique identifier for the regulatory clause
section_title	Title of the regulatory clause
standard_text	Full textual content of the clause
source_standard	Source document name (e.g., EN 1175, LVD)
is_binding	Boolean indicating if the clause is legally binding
MDW_single	Matched domain-specific words (tokens)
MDW_complete	Matched complete domain-specific phrases
PST	Matched internal product description section
max_similarity	Maximum cosine similarity score from semantic retrieval
z_score	Normalized similarity z-score for certainty classification
certainty_label	Categorization label (e.g., Very Certain Relevant)
answer	LLM-generated label and conclusion (text)
RD	Binary decision derived from model output: 1 if the requirement is applicable, 0 otherwise (i.e., both "Requirement Not Applicable" and "Not a Requirement")
prediction	Processed model output in binary form (used in evaluation)
label	Ground-truth label for validation (1 = Applicable, 0 = Not)

4.6.1 System Overview and Architecture

The solution is delivered as a multipage web application developed in Streamlit, hosted within the Databricks (Azure) environment, with each page corresponding to a different stage in the compliance pipeline. The workflow aligns with the methodology introduced in Sections 4.3 through 4.5, and is implemented as follows:

1. **Product Summary Upload:** Accepts internal XML documentation and extracts product-level content.

2. **Standard Upload:** Processes known regulatory files and maps them using a persistent library_map.json. A prepopulated standards database reduces user overhead: if the standard already exists in the system, only the target description (TD) must be uploaded. The system supports uploading and analysing multiple standards simultaneously, allowing for multi-document compliance screening and aggregation of results across regulations.

3. **Semantic Analysis:** Performs embedding, similarity matching (as per Section 4.4.3), and domain keyword detection.

4. **LLM Classification:** For each high similarity match, invokes a hosted LLM endpoint to generate final compliance assessments (as per Section 4.5).

5. **Visualization:** Displays aggregated results including scores, labels, and traceable clause metadata.

6. **Human-in-the-Loop (HITL) Review:** Allows expert reviewers to inspect, override, and annotate the final LLM outputs.

To simplify and increase usability, the system supports a one-click automation mode: if the user uploads only a target description (TD) that includes metadata on relevant standards, the system automatically executes all upstream stages, from standard retrieval and embedding to clause matching and classification, until it pauses at the HITL review interface.

4.6.2 Traceability and Data Integrity

At each stage, source identifiers such as `section_id`, `sid`, and `ancestor` are stored to enable full back-tracing of decisions. The aggregate sections of Section 4.4.4 are also stored with their source to preserve traceability. The application manages the intermediate state using `st.session_state`, allowing users to navigate out-of-order steps while maintaining data consistency.

File mappings are tracked using a structured index (`library_map.json`), which ties each uploaded standard to a canonical internal name. This ensures that reanalysis can occur on identical versions of a regulation, supporting repeatability.

4.6.3 Human-in-the-Loop Review and Interaction

The final step of the tool introduces a controlled Human-in-the-Loop (HITL) interface. Here, users are presented with the LLMs classification results from Section 4.5, including clause metadata, product evidence, similarity scores, and natural language justifications.

Users can:

- override the predicted certainty label;
- edit or append justification text;
- annotate errors or flag ambiguous cases;
- export selected rows to SystemWeaver for downstream system design.

All errors provided by experts made in the HITL interface are stored in an internal database. These edits are linked to the original clause ID and time-stamped, allowing downstream systems to retrieve historical decisions, perform compliance audits, or bootstrap future supervised training datasets.

This interactive page serves two purposes: (1) it allows for expert verification of AI-generated results in legally sensitive contexts, and (2) it generates annotated data that could inform future fine-tuning efforts. Any modifications made during the HITL review are stored alongside the original predictions for full traceability.

4.6.4 Deployment Context and Access

The complete system is deployed in the secure Azure environment of Volvo Penta. All document processing and inference occur within the corporate tenant, no sensitive files or embeddings are sent externally. The LLM classification endpoint is deployed as an Azure Function and is accessed via token-restricted API calls.

The final outputs can be downloaded as CSV or Excel files or programmatically submitted to internal compliance tools. In particular, the complete, clause-level analysed version of the standard, including all predicted labels, supporting evidence, and expert edits can be uploaded to SystemWeaver via its official API endpoint. This integration ensures that compliance results are preserved alongside formal system documentation, enabling cross-functional traceability. This setup ensures that the research pipeline is usable within operational constraints and aligned with Volvo Penta's compliance workflow.

4.7 Validation Methodology

This chapter outlines how the proposed compliance pipeline was evaluated both quantitatively, using expert-labelled ground-truth data, and qualitatively, through targeted stakeholder interviews.

4.7.1 Quantitative Evaluation Design

Dataset

The primary benchmark consists of 256 clause–project pairs, each manually derived by the conclusions of the senior compliance engineers. The pairs are the text or body of one section of the standard, and then the conclusion for that section mapped to a (1/0) label. These pairs originate from a full analysis of the EN1175:2020 standard, as introduced in Section 4.2. This ground-truth expert file provides a reliable basis for evaluating classification accuracy.

- **Source:** Expert-labelled compliance file.
- **Scope:** 256 clause–project pairs.
- **Labels:** 'Applicable' and 'Not Applicable', mapped to binary targets (1/0).

The complete structure of the model output processed for evaluation is described in Section 4.5.5. This schema includes key fields such as the original clause text, product section match, LLM-generated classification, and the binary decision flag

`requirement_decision`, which is used as the predicted label of the model during validation.

To compute quantitative evaluation metrics such as accuracy, precision, and recall, the model output was originally in one of three categories: Requirement applicable, Requirement not applicable, and Not a requirement was mapped to a binary format. Specifically, only the applicable requirement was treated as a positive prediction (label = 1), while the requirement not applicable and the requirement not required were assigned to negative predictions (label = 0). This binary mapping reflects the core objective of identifying clauses that are truly relevant and applicable to the project and aligns with how the ground-truth data were annotated (binary format: applicable/not applicable).

Metrics

The system's classification output will be compared against the ground truth using the following metrics:

- **Accuracy** — fraction of correct predictions.
- **Precision** — fraction of predicted positives that were actually correct.
- **Recall** — fraction of actual positives that were correctly predicted.
- **F_2 Score** — harmonic mean of precision and recall, weighted to favour recall.

These metrics are selected to reflect both the correctness and completeness of predicted applicability labels, especially favouring recall due to the cost of false negatives in compliance contexts. The F_2 score is chosen over F_1 to place greater emphasis on recall, as missing a truly applicable requirement poses a higher risk than misclassifying a non-applicable one. In case of borderline predictions or misclassification, the human-in-the-loop mechanism (Section 4.6) provides a critical safety net for review and correction.

A total of 16 combinations were tested on the validation set, consisting of four embedding models mentioned in Section 4.4.1 and four LLMs mentioned in Section 4.5.4. Furthermore, due to the non-deterministic behaviour of LLMs[84], even with a lower temperature setting in the experiment, the model may still generate different outputs each time. To assess the impact of non-deterministic behaviour on this task, we selected the combination with the highest recall score identified in the comparative evaluation and run it on the validation set several times using identical input prompts in order to observe variations in the generated answers. Wang et al. conducted 10 repeat runs in their self-consistency study[85]. Their research also suggested that five or ten repetitions are generally appropriate in practice, considering computational cost. In this study, we conducted five repeat runs as a preliminary assessment of the consistency of the model.

4.7.2 Qualitative Evaluation Design

To assess the usability, interpretability and decision support potential of the prototype, we conducted semi-structured interviews with industry professionals. This method is well suited for the evaluation of interactive systems, as it combines the consistency of a guided structure with the flexibility to explore emerging issues in depth [86, 87]. Such interviews are widely used in human-computer interaction and software engineering to gather user-centred and rich feedback [86].

Participants

We recruited six participants: 2 **Compliance Engineers**, 2 **System Engineers**, and 2 **Project Managers**. This sample size aligns with usability research suggesting that five to six participants can uncover the majority of usability concerns [88], while also approaching thematic saturation in qualitative studies [89].

Participants were selected using a maximum variation sampling strategy to ensure representation in key roles and capture a wide range of perspectives [87]. Furthermore, following expert sampling principles, all participants had a minimum of three years of experience or had participated in at least one full product development lifecycle. This criterion ensured that interviewees could provide informed, experience-based evaluations of the applicability of the prototype in the real world.

An overview of the roles and experience of the participants is provided in Tables 4.5 and 4.6.

Table 4.5: Interview participants: IDs and roles.

ID	Role
P1	Chief Project Manager
P2	Regulatory Affairs Manager
P3	System Engineer
P4	Chief Project Manager
P5	Senior Safety Compliance Engineer
P6	Specialist Systems Engineering Leader

Interview Protocol

Each interview lasted 45–60 minutes and followed a semi-structured guide:

1. **Warm-up:** Establish the role and experience of the participant with compliance evaluation.

2. **Prototype Walkthrough:** Present representative outputs including various label combinations (e.g., Applicable, Not Applicable, No Requirement) and confidence levels (Very Certain, Possibly Relevant, etc.).

Table 4.6: Interview participants: organizational focus and experience.

ID	Organizational Focus	Experience
P1	Platform development and compliance execution	25 years total, 15 as Project Manager Engineer (PME)
P2	Industrial regulatory compliance (non-marine diesel, electromagnetics (EM))	3 years, specialized in EM
P3	Functional architecture, components, and charging systems	8 years at company
P4	Platform lifecycle and planning	3 years at company
P5	Application of EU directives and standards (Conformité Européenne (CE), EMC)	10 years, extensive with EU compliance
P6	Embedded system requirements and diagnostics	5 years in system-level engineering at company

3. **Guided Discussion:** Explore perceptions of label accuracy, interpretability, and trustworthiness using a structured nine-question protocol.

4. **Wrap-up:** Elicit feedback on integration potential and desired improvements.

The complete interview protocol is provided in the Appendix G. The thesis supervisor reviewed and validated the content to ensure relevance, clarity, and alignment of the content with the research objectives. Feedback from this review was incorporated into the final set of questions.

Participants were shown model outputs from the same industrial case as in 4.2.2, but across two types of regulatory sources: the EN 1175 standard, which had been manually analysed and labelled in advance, and three EU directives, which lacked any predefined classifications. This setup allowed participants to evaluate the historical accuracy of the model using the labelled data, while also evaluating its interpretability and potential usefulness on previously unseen inputs. Details of these regulatory sources are provided in section 4.2.

All interviews were conducted one on one, recorded with participant consent, transcribed, and anonymised prior to analysis. Although the discussion followed a structured protocol, the interviewers remained flexible, asking follow-up questions or exploring additional topics where relevant, in accordance with the semi-structured format.

Analysis Approach

We applied thematic analysis following the Braun and Clarke six-phase framework [90], which is widely used in qualitative research to identify and analyse meaning patterns in data. The process involved:

1. **Familiarization:** Transcripts were read multiple times to gain an overall sense of the data.

2. **Initial Coding:** Salient segments were systematically tagged using open coding. Each meaningful unit (e.g., comment, reflection, concern) was labelled with short descriptive codes.

3. **Theme Development:** Codes were iteratively grouped into candidate themes that captured shared meanings or concerns.

4. **Theme Review:** The initial themes were refined by reviewing them against the entire dataset to ensure internal consistency and distinctiveness.

5. **Defining and Naming Themes:** Each theme was clearly defined, named, and assigned to the research focus, particularly regarding the usability, trust and potential of the system.

6. **Reporting:** Themes are presented in Section 5.2, supported by representative quotes from the participants.

This approach, also known as semantic coding when focused on explicit content, is widely recognised in empirical software engineering as an effective means of translating qualitative data into actionable findings [91]. Themes such as Interpretability Challenges and Integration Concerns emerged from repeated participant comments, and these form the foundation of our qualitative results presented in Section 5.2.

By integrating structured analysis with the perspectives of domain experts, this qualitative evaluation complements our predictive validation and offers a comprehensive view of the readiness of the system for real-world deployment.

Coding procedure

Using an **inductive, data-driven** strategy, both authors first open-coded two "pilot" transcripts in parallel (two 90-minute sessions). This yielded roughly 56 provisional codes. A 60-minute consensus meeting then aligned code names and definitions, creating a shared mini-codebook. With that common frame in place, the remaining four transcripts were divided between the authors and coded separately, followed by a second 60-minute reconciliation session. During this stage the 56 codes were merged into conceptually related *code clusters* and iteratively refined into the six final themes reported in Section 5.2.

5

Results

5.1 Quantitative Evaluation Results

5.1.1 Evaluation Matrix

Based on the above verification method in Section 4.7.1, we tested 16 combinations on the validation set. The user input part extracts different matching project descriptions depending on the embedding model used. In actual deployment, we get access to the two OpenAI LLMs, gpt-35-turbo and gpt-4o-mini, via the Azure OpenAI platform. The Meta-llama-3.3-70B-Instruct and Claude 3.7 Sonnet are deployed on databricks and can be used directly by calling their respective endpoints. The full evaluation set is shown in table 5.1.

Table 5.1: Evaluation matrix of 16 combinations of model on the validation set.

	GPT-4o-mini	**GPT-3.5-turbo**	**Meta-llama-3-3-70b-instruct**	**Claude 3.7 Sonnet**
Legal Bert	Accuracy: 0.715 Precision: 0.769 Recall: 0.674 **F2 Score: 0.691**	Accuracy: 0.707 Precision: 0.676 Recall: 0.877 **F2 Score: 0.828**	Accuracy: 0.789 Precision: 0.792 Recall: 0.826 **F2 Score: 0.819**	Accuracy: 0.789 Precision: 0.739 Recall: 0.942 **F2 Score: 0.893**
BGE-m3	Accuracy: 0.738 Precision: 0.771 Recall: 0.732 **F2 Score: 0.739**	Accuracy: 0.719 Precision: 0.679 Recall: 0.906 **F2 Score: 0.849**	Accuracy: 0.793 Precision: 0.774 Recall: 0.870 **F2 Score: 0.849**	Accuracy: 0.777 Precision: 0.726 Recall: 0.942 **F2 Score: 0.889**
text-embedding-ada-002	Accuracy: 0.723 Precision: 0.760 Recall: 0.710 **F2 Score: 0.720**	Accuracy: 0.703 Precision: 0.661 Recall: 0.920 **F2 Score: 0.853**	Accuracy: 0.773 Precision: 0.767 Recall: 0.833 **F2 Score: 0.819**	Accuracy: 0.785 Precision: 0.732 Recall: 0.949 **F2 Score: 0.896**
databricks-gte-large-en	Accuracy: 0.746 Precision: 0.774 Recall: 0.746 **F2 Score: 0.752**	Accuracy: 0.668 Precision: 0.635 Recall: 0.906 **F2 Score: 0.834**	Accuracy: 0.766 Precision: 0.760 Recall: 0.826 **F2 Score: 0.812**	Accuracy: 0.777 Precision: 0.724 Recall: 0.949 **F2 Score: 0.894**

5.1.2 Average Performance Over Multiple Runs

From the above results, it is evident that Claude 3.7 Sonnet consistently achieves significantly higher recall and F2 scores across all four embedding models compared to the other language models. Among them, the combination with text-embedding-ada-002 yields the highest recall and F2 score overall. Therefore, we selected the text-embedding-ada-002 + Claude 3.7 Sonnet combination for further evaluation and repeated the test five times using the same setup. The results are presented in

the following table 5.2.

Table 5.2: Evaluation metrics of the model using text-embedding-ada-002 + Claude 3.7 Sonnet on validation set over five repeated tests.

	Test 1	**Test 2**	**Test 3**	**Test 4**	**Test 5**
Accuracy	0.785	0.773	0.773	0.770	0.773
Precision	0.732	0.720	0.725	0.718	0.722
Recall	0.949	0.949	0.935	0.942	0.942
F2 Score	0.896	0.892	0.884	0.887	0.888

Table 5.3: Mean and standard deviation of evaluation metrics over five repeated validation tests (text-embedding-ada-002 + Claude 3.7 Sonnet).

Metric	**Mean**	**SD**
Accuracy	0.775	0.006
Precision	0.723	0.005
Recall	0.943	0.006
F_2 Score	0.889	0.005

Table 5.2 details the metric scores for each of the five validation runs. Based on these results, Table 5.3 consolidates the model performance by reporting the mean and standard deviation ($n = 5$) for each metric, giving a clear picture of both its predictive capacity and the variability between runs.

Table 5.4: Average evaluation metrics of the model using text-embedding-ada-002 + Claude 3.7 Sonnet.

Metric	**Average Score**
Accuracy	0.775
Precision	0.723
Recall	0.943
F_2 Score	0.889

5.1.3 Representative Confusion Matrix

The confusion matrix shown in Figure 5.1 corresponds to Test 1, one of the five validation runs performed during model evaluation. Although performance metrics were averaged for all runs (see Table 5.2) to assess overall consistency and robustness of the model, this specific matrix is presented as a representative example to illustrate the behaviour of the model in individual predictions.

Due to the high similarity observed across all five confusion matrices, both in structure and in class distribution. We chose to include only this one in the main body of the thesis for clarity and conciseness. The remaining four confusion matrices, corresponding to Tests 2 through 5, are provided in Appendix E for reference.

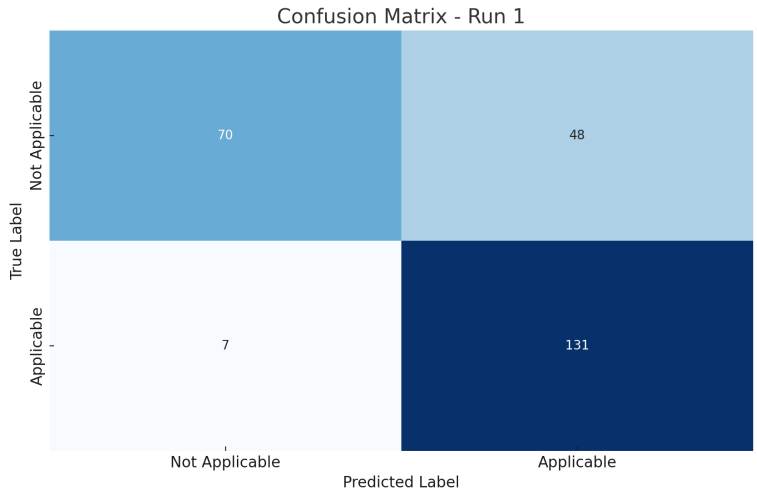

Figure 5.1: Confusion matrix for `text-embedding-ada-002` + Claude 3.7 Sonnet on Validation Set (Test 1).

5.2 Qualitative Evaluation Results

Thematic analysis of the six interview transcripts produced six primary themes regarding the perceptions of participants about the precision, usability, risks, and potential integration of the prototype into real-world workflows. These themes emerged across the three participant roles described in 4.7.2 and reflect shared patterns as well as role-specific concerns. Each theme is described in the following with representative anonymised quotes.

5.2.1 Theme 1: Conditional Trust in Automated Outputs

All six participants said they would *use* the model, but only with human oversight. P1 put it bluntly, "I would definitely try to use this as much as possible … but I wouldn't trust it 100 %." P2 framed the stakes in regulatory terms: "Everything is my risk … If the model makes a mistake and we miss a critical requirement, it could be a disaster." Reinforcing that caution, P4 noted that "you still need to go through the data anyway," adding that a 77 % hit rate is acceptable only if "we don't miss anything critical outside that."

5.2.2 Theme 2: Perception of Time Savings—and the Risk of Double Work

Five of the six interviewees (P1, P3, P4, P5, P6) saw clear time-saving potential during early workshops. P3 recalled spending "several weeks with multiple departments" and remarked that "if this tool can reduce that, it's huge." Likewise, P5 estimated that it "could save 30 % of the time … maybe even more if it's accurate enough." However, the same people warned that benefits disappear if every

clause still needs verification; as P1 put it, "then it's not really saving time—it's double work."

5.2.3 Theme 3: Risk of Missing Critical Requirements

All six participants identified false negatives as the single most serious risk. P2 stressed that "even missing one critical requirement could mean the product is not compliant," while P4 argued that if something is flagged "*No match* and it turns out to be relevant ... the whole filtering approach breaks." P6 summed up the worst-case scenario: "We ship something that turns out to be non-compliant because of a missed label."

5.2.4 Theme 4: Integration Potential and Feature Suggestions

Three participants (P1, P3, P6) proposed concrete feature requests to aid adoption. P3 wanted functional-safety clauses highlighted "in a separate list" because they are "hard to reverse-engineer if missed early." P1 asked for "a visual summary—how many are applicable or uncertain—plus a time estimate to finish the uncertain ones," and P3 added that direct export to systems-engineering tools "would save time" by preserving traceability IDs.

5.2.5 Theme 5: Quality of Model Explanations and Output

Three participants (P3, P5, P6) praised the clarity of the generated text. P3 admitted that "some of the conclusions written by the model were actually better than the ones I had written myself," and P5 found that the structured output "definitely makes it easier to understand" than reading raw standards documents.

5.2.6 Theme 6: Expectations for Model Adaptability and Learning

All six participants expected the model to evolve over time. P6 said that "if it learns from our input and adapts, then we can trust it more," while P2 warned that "standards change, so if the tool can't evolve, it becomes outdated fast." P4 hoped the system would "remember how we judged similar clauses before ... and adjust next time."

Summary

These results reveal a cautiously optimistic attitude towards the prototype. All participants appreciated its potential to reduce manual effort, improve clarity, and structure their work, but emphasised the need for safeguards, adaptability, and thoughtful integration. Successful adoption depends on sustained model accuracy, transparency, and responsiveness to real-world changes.

5.3 Prototype Output Used in Evaluation

The prototype interface was central to the evaluation process. Participants interacted directly with each stage of the pipeline, from product upload and clause matching to model predictions and human-in-the-loop (HITL) review.

Figure 5.2 shows the clause-level prediction and review interface, which was used to verify labels, inspect similarity-based justifications, and override model outputs.

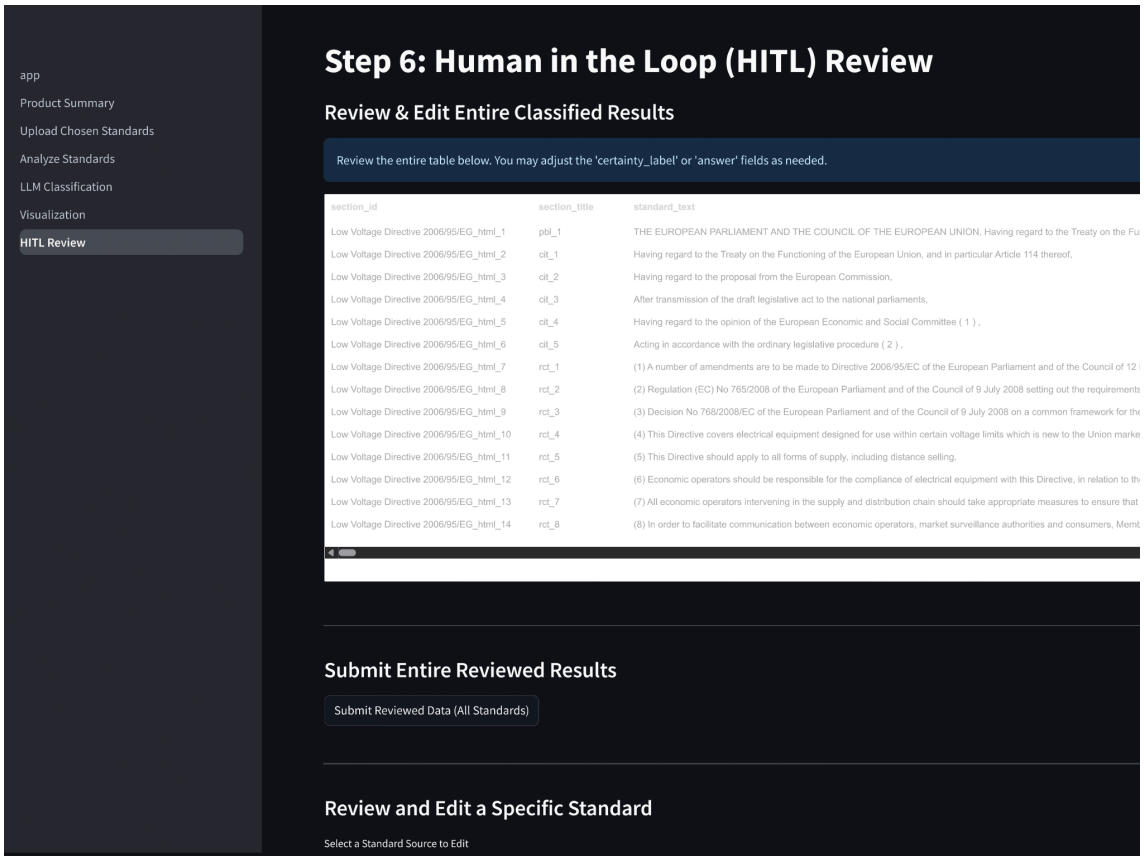

Figure 5.2: Final HITL review interface displaying the `section_id`, `section_title`, and `standard_text` columns. Additional fields are available in the scrollable view to the right.

Additional screenshots covering the six pipeline stages are included in Appendix F.

5.3.1 Example Output Row from Prototype

Table 5.5 shows a full example of a final output row from the prototype system. It contains the original clause text, the engineering content matched, the embedding scores, the matched domain terms, the LLM-generated justification, and the final classification label. This row is representative of the 515 clauses processed by the system, with the remaining outputs following a similar structure.

Table 5.5: Example row output from the prototype's final classification pipeline.

Clause ID	EN 1175:2020 – 4.3.1
Clause Title	Low voltage/high voltage
Standard Text	Safety shall not be compromised at any voltage level that can occur. Electrical systems of trucks powered by lead-acid batteries shall be designed so that all functions operate in the voltage range from 70% up to 120% of the nominal battery voltage. These limits shall be adapted to other energy sources technologies by the manufacturer.
Source Standard	EN 1175
Binding Clause	True
Matched Domain Words (Single)	power, system, source, battery, voltage, energy, limit
Matched Domain Words (Complete)	voltage, over voltage, Power, under voltage
Matched Product Section Text	*[Truncated: internal engineering description covering regulatory compliance, battery types, safety standards, voltage constraints, and lifetime expectations]*
Similarity Score	0.859
Z-score	1.268
Certainty Label	Very Certain Relevant
LLM Final Label	**Requirement Applicable**
LLM Justification (Excerpt)	*"This section is a requirement because it uses prescriptive language with 'shall' statements... The clause aligns with project data showing battery systems and voltage control, making it directly applicable."* See Appendix H for the complete justification.

6

Discussion

This chapter interprets the findings from both quantitative and qualitative evaluations in relation to the research objectives. The results are discussed in the context of the three research questions presented in Chapter 1:

- **RQ1:** *How can we develop a system that automatically links regulatory requirements to relevant product and engineering information in Volvo Penta's lifecycle?*
- **RQ2:** *How can this system be effectively integrated into Volvo Penta's engineering workflows to support traceability, usability, and transparency in decision-making?*
- **RQ3:** *How does the system perform in terms of accuracy, interpretability, and explainability, and to what extent does it support decision-making based on expert evaluations and real-world scenarios?*

The quantitative results provide a direct response to RQ1 and RQ3 by assessing the technical performance of the model across key compliance metrics. At the same time, by quantitatively analyzing model performance aspects such as running time and consistency in model output, the usability of the model mentioned in RQ2 is also emphasized. In addition, the qualitative themes primarily inform RQ2 and the interpretability dimension of RQ3.

In addition to the analysis of the results of the model, this chapter also discusses the limitations, validity, and directions of the model for future work.

6.1 Discussion Regarding the Quantitative Evaluation

This section will quantitatively analyse and discuss the model based on the results of the experiment in Section 5.1. This includes a comparative analysis of the performance of the model itself, a analysis of the evaluation matrix of the model output results, and a statistical analysis and discussion of the uncertainty of the model output.

6.1.1 Model Evaluation Metrics Analysis

The evaluation of the final model configuration, combining `text-embedding-ada-002` for semantic retrieval with Claude 3.7 Sonnet for clause classification, shows promising performance across multiple compliance-related metrics. The model provided stable and accurate predictions on five independent runs.

Overall Performance Summary

Averaged over five cross-validated runs, our system achieves **77.5 % accuracy**, **72.3 % precision**, **94.3 % recall**, and a recall-weighted $\mathbf{F_2}$ of **88.9 %**. Clause-level benchmarks in the literature range from 79 % to 89 % accuracy. Moon et al. [67] report 88.9 % accuracy with $F_1 = 0.93$ on construction-risk clauses; Amaral *et al.* [13] reach 84.6 % accuracy but only 82.4 % recall on GDPR data-processing agreements; and Braun & Matthes [68] obtain ≈83 % accuracy with $F_1 = 0.88$ on multilingual terms and conditions clauses. The precision of our model is slightly lower than these baselines, but its significantly higher recall raises $\mathbf{F_2}$ above all three, which is crucial for compliance tasks where missing a relevant clause is costlier than a false positive (see 6.2). We also surpass the GPT-4 GDPR checker of Hassani *et al.* [70], which reports 81 % accuracy and $F_1 > 0.80$ but does not optimise for recall.

Recall-Oriented Strategy and Justification

The model consistently prioritised recall across all runs, achieving 94% in each. This design aligns with compliance objectives, where failing to detect an applicable requirement (a false negative) can pose significant legal and safety risks. By prioritising completeness over strict precision, the system ensures that potentially relevant clauses are flagged for further inspection, even if some eventually prove inapplicable.

Precision Trade-offs and Interpretation

The precision of 72.3% reflects the proportion of correctly identified applicable clauses among all those predicted as applicable. Although lower than recall, this is a reasonable trade-off considering the classification complexity and variability in regulatory language. False positives are often due to nuanced clauses or lack of complete project information in the input text. Such errors are manageable in practice, particularly in systems that incorporate human review of outputs.

Run-to-Run Stability

The performance of the model was consistent between the five validation runs (Table 5.2), with only slight variations between the metrics. The accuracy ranged from 77.0% to 78.5%, and the F_2 score varied between 0.884 and 0.896. This indicates that prompt format, retrieval strategy, and temperature setting (`temperature = 0.2`) contributed to robust and reproducible behaviour in repeated evaluation.

Confusion Matrix Analysis

The confusion matrix in Figure 5.1, derived from Test 1, provides insight into the classification tendencies of the model. Of 256 validation samples, 131 were correctly classified as applicable (true positives), while 7 were missed (false negatives). The model also flagged 48 false positives and correctly rejected 77 clauses as not applicable or not a requirement.

This matrix confirms the recall-oriented bias of the model. Only a small number of applicable clauses were missed, while more were over-included. In compliance workflows, this over-inclusiveness is preferable, as it avoids the risk of overlooking valid regulatory obligations.

Analysis of Misclassification Cases

After reviewing the 55 misclassified cases shown in Figure 5.1 in Test 1, the reasons are summarised as follows.

- The 48 false positives were caused by incorrect attribution of responsibility or application scope. For example, some clauses are the responsibility of the Original Equipment Manufacturer (OEM), and the subject of some regulatory clauses is not included in the product scope of Penta. The reason for the misclassification is the lack of sufficient project information. Although the model is provided with project background and some project-specific keywords, it is not enough to cover all components and technical details. Therefore, it is essential to establish a specific compliance knowledge library. The purpose of saving the results of the manual review is to support the subsequent establishment of a compliance knowledge base.

- Among the 7 False Negative cases, 2 were caused by the model's failure to correctly identify whether the clause constitutes a requirement. This could potentially be addressed by providing clearer definitions of a requirement, and such errors are considered within an acceptable margin of error. The remaining false cases were due to the model's difficulty in determining whether the regulatory subject is within the product scope of Penta, which is the same as the cause of the error of False Positive cases.

The analysis of misclassification cases also highlights the model's emphasis on the interpretability and traceability mentioned in RQ2 and RQ3. When experts review the model's output, they can clearly see its reasoning process and the supporting evidence, allowing them to assess whether the reasoning is reasonable.

Impact of Limited Data Availability

Although, the experiment demonstrates that even with limited resources, LLM-based pipelines can offer useful and scalable support for compliance interpretation tasks. It is important to note that the model was validated using only a single annotated engineering project with 256 total samples. Despite this limited dataset,

the results remain encouraging. The model generalises well across repeated tests. However, broader validation using projects from other domains or industries will be necessary to fully assess its generalisability and readiness for practical deployment.

6.1.2 Model Performance Analysis

Running Time

The running time of the model plays a crucial impact on the user experience. In the method proposed in this study, the running time of the large language model accounts for a large proportion of the total running time, and it varies greatly depending on the LLM selection. The table 6.1 below summarises the approximate run-times of the four selected LLMs over a total of 256 prompts in the validation set. Since the change in running time is affected by many factors, such as network speed, service resource load, etc., this table describes the approximate time the model runs on the validation set.

Table 6.1: Running time of different LLMs on the validation set.

	GPT-3.5-turbo	**GPT-4o-mini**	**Meta-llama-3-3-70b-instruct**	**Claude 3.7 Sonnet**
Approximate Running Time	1 h 20 mins	20 mins	30 mins	45 mins

Among the four models evaluated, the GPT-3.5-turbo exhibited the longest runtime, with a single regulatory clause that took nearly one and a half hours to process. In contrast, GPT-4o-mini demonstrated significant improvements in speed, offering a much more lightweight and efficient solution. Meta-llama-3-70B-Instruct showed slightly slower performance compared to GPT-4o-mini, but remained within a relatively fast range. Claude 3.7 Sonnet, while slower than Meta-llama-3-70B-Instruct, also maintained an acceptable runtime overall. As one system engineer pointed out in the interview (Section 5.2.2) that compared with traditional methods, such as when conducting compliance work with EN 1175, which is the same standard used in our validation set, where several days were spent on workshops followed by several days for decision-making, 40-minute running time of the model is entirely acceptable. Even longer durations are considered reasonable, as they form part of the necessary preparation process. In view of this, the time taken by these four models to complete this task is considered reasonable.

Closed- vs Open-Source Model Trade-offs

An interesting observation arises when we compare the closed baseline (Claude 3.7 Sonnet) with the strongest open-source alternative (Meta Llama-3-70B-Instruct) in Table 5.1. Across all four embedding models, Claude leads by roughly one to two percentage points in F_2 and by about 10 pp in recall (e.g. 94.9 % vs. 83.3 % with `text-embedding-ada-002`). However, Table 6.1 shows that Llama runs the same 256-clause workload in 30 minutes, a full third faster than Claude's 45 minutes, and, being open source, maintains all regulatory texts on premise, allows exact

reproduction of results, and eliminates perpetual API fees once GPUs are amortised. In the short term, Claude's slightly higher recall is helpful during rapid prototyping; in the long run, however, the governance, cost, and flexibility advantages of Llama outweigh its narrow performance gap. A pragmatic strategy is therefore to start projects with the closed model for speed, then transition to (or run it alongside) the open model once internal fine-tuning closes the remaining recall margin.

Consistency in Model Output Format

In the experimental instructions, the LLMs were guided to complete the task in three distinct steps, the first step requiring the model to select one answer from three predefined options. However, the four LLMs show varying levels of adherence to the instructions and their consistency in output formatting also differed. The figure below presents the distribution of the selected options generated by 4 LLMs on the validation set using the same embedding model "text-embedding-ada-002". Additionally, the distribution of output from the four LLMs are summarised as Figure 6.1:

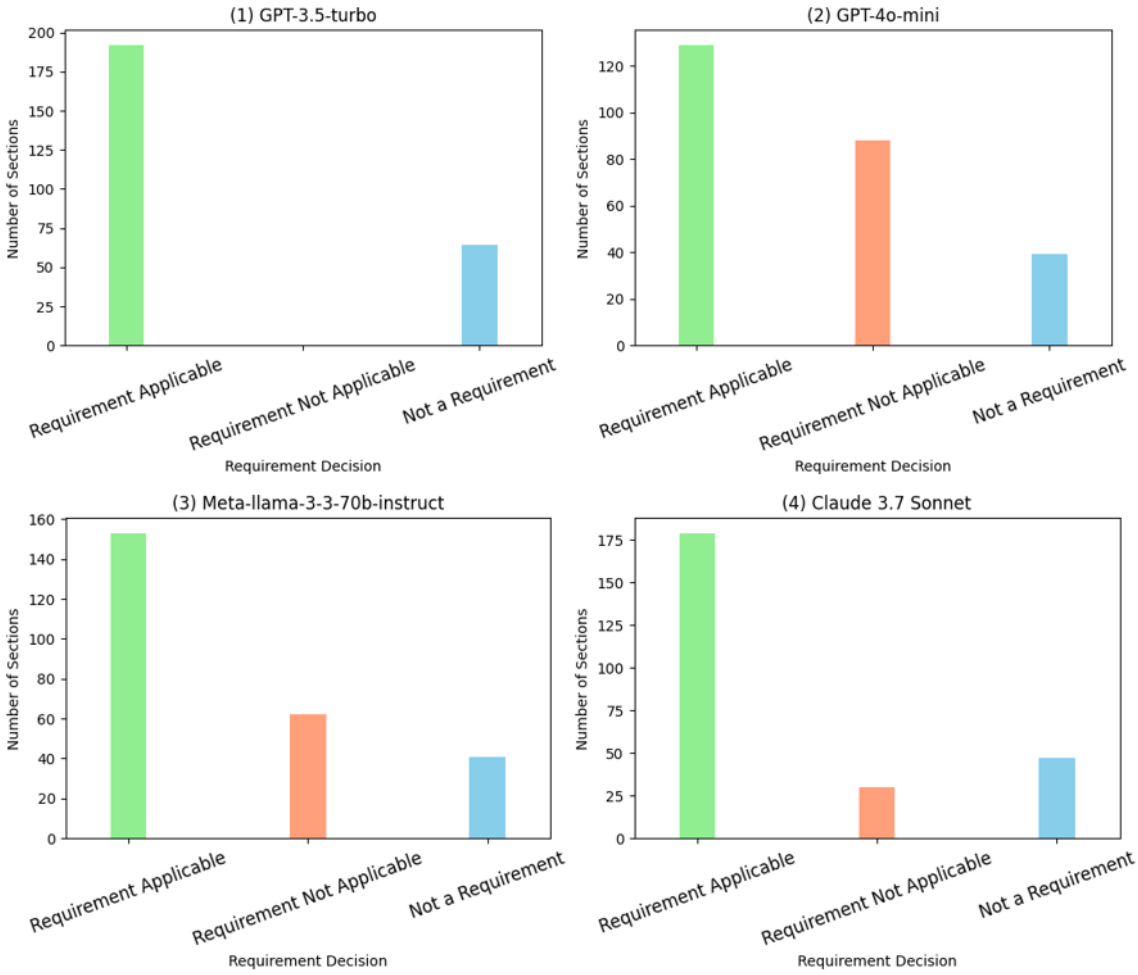

Figure 6.1: The distribution of the answers from the four LLMs.

- As shown in the figure above, GPT-3.5-turbo fails to correctly identify the

"Not a Requirement" category. In terms of output format, for each output, it can strictly follow the 1, 2, and 3 parts to organise the answers. However, there is inconsistency in the answer to the first part. Although some outputs begin directly with "1. Requirement Applicable", others start with "1. Choose one of the three answers: Requirement Applicable".

- GPT-4o-mini can successfully choose the answers from the three predefined options according to the instructions. Its outputs consistently follow the 1-2-3 structure, and the format is uniform across all responses. Specifically, point 1 is always presented in the format "1. Requirement Applicable" (or the respective chosen label).

- Meta-llama-3-3-70b-instruct also successfully completed the answer selection according to the instructions. Its output format of results also strictly follows the three steps. However, a small number of responses began with introductory phrases such as: "Based on the provided information, I conclude that:"

- Claude 3.7 Sonnet successfully identified the answers in accordance with the instructions. From the output format, it also strictly followed the instructions to output the three-paragraph conclusion, but the format was not uniform. Some answers were structured according to 1, 2, 3 points, and some answers were arranged according to "Assessment", "Reasoning Process", "Subject and Requirements".

From the perspective of output format consistency of the above models, GPT-4o-mini and Meta-llama-3-3-70b-instruct can generate answers very well according to the format requirements of the instructions and are able to maintain a consistent structure across different outputs. GPT-3.5-turbo struggles to accurately follow the instructions for answer selection, but can roughly output answers with consistent formats. Claude 3.7 Sonnet follows the instructions well in terms of content, but the output format shows some variability.

6.1.3 Evaluation of Non-Deterministic Behaviour of LLMs in Compliance Requirements Generation Tasks

Consistency Analysis

We performed a consistency analysis on the results of the five repeated tests in Section 5.1. For each clause in the validation set, we found the mode label (i.e., the label that occurs most frequently) across the five test results and calculated the proportion of tests that equalled this mode label. The number of clauses at different levels of agreement ratio is shown in Table 6.2.

Table 6.2: Clause count by agreement ratio.

Agreement Ratio	0.6	0.8	1.0
Clause Count	6	12	238

As can be seen in the table, the agreement ratio of most clauses is 100%, and very few are 80% and 60%. For clauses with an agreement ratio greater than 70%, which means that the model produced consistent results in at least four of five tests, we consider the model to have a high consistency in the answer to this clause. According to the statistical results, the proportion of clauses with high consistency reaches 97.66%, indicating that the model has high consistency in the performance of the compliance requirements generation task.

Analysis of Inconsistent Samples

Six samples with an agreement rate below 70% were analysed. The differences in the model outputs seem to come from the following reasons:

- **Not enough detail about how the product is used.** In these six samples, the main issue was whether the requirement was part of Penta's area of work. The prompt only gave a general overview of the company's business, project background, and field of use, but did not include specific details about the parts involved in the project. Because of this, the models gave different answers depending on how they understood the situation. In all five tests, the models used the prompt information correctly, but came to different conclusions because they interpreted the information in different ways. For example, when deciding whether the emergency brake device requirement was applied, some responses said it did not fit the business scope, while others focused on how the parts work together and said it was relevant.

- **Different focus on the prompt information.** Even though the input was the same, some answers paid more attention to the scope of use, while others focused on whether the project description matched. This difference in focus led to different ways of thinking and different results.

The above analysis suggests that providing sufficient technical details of the product can reduce the inconsistency of the responses. However, since this compliance requirement is generated in the early stages of product development, it is difficult to obtain enough product details. Furthermore, the TD is manually created and the quality of the document varies each time, which can also affect the product details. In this case, the model should tend to identify the clause as applicable, leaving the final determination to domain experts. Regarding the prompt information focus issue, it is difficult to define which part should be emphasised for a given clause, which also requires experts to further confirm based on their experience. By integrating this model into the existing workflow, experts can modify and save the final output results. These results can then serve as high-quality training data, helping to improve the accuracy of future predictions through machine learning techniques.

6.1.4 Summary

The model shows good accuracy, with particularly strong recall and F2 scores, which addressed RQ3 and indicating its reliability in minimizing the risk of missing any

applicable requirements. The results also demonstrate its ability to align regulatory requirements with relevant product and engineering information, which is the central goal of RQ1. At the same time, the model has efficient running time and high consistency of answers, which proves that the model has good usability, corresponding to RQ2. In addition, its traceability and interpretability allow for a detailed analysis of misclassification and inconsistent cases.

6.2 Discussion Regarding the Qualitative Evaluation

This section interprets the six qualitative themes identified in Section 5.2 and evaluates their implications for trust, usability, and future adoption of the prototype. These findings primarily address RQ2 by evaluating workflow fit, human-AI collaboration, and system usability, while also informing RQ3 on interpretability and human-centred trust.

6.2.1 Theme 1: Conditional Trust Reflects Real-World Accountability

Theme 1 revealed that participants were cautiously optimistic but ultimately unwilling to delegate the final responsibility to the model. Although they appreciated the structure and predictions of the model, they saw it primarily as a support tool, not a replacement for expert judgment. This reflects a form of institutional trust, where the key concern is who bears the risk if something goes wrong. As one participant put it, 'Everything is my risk'. Even with technically sound performance, trust is withheld unless organisational safeguards and accountability structures are in place. This aligns with patterns in the interaction between humans and AI where transparency, control, and clear responsibility are essential in high-stakes contexts [92].

A related concept closely related to trust is transparency, which is formally defined in Section 2.7. In this study, transparency was improved by the structured outputs and explanation fields provided by the prototype, such as the similarity score, keyword matches, and the model reasoning text. These elements allowed participants to trace the origin of each conclusion and verify the logic behind it. Even when participants disagreed with a model decision, they acknowledged that being able to 'see the reasoning' increased their confidence in using the tool responsibly. Thus, transparency plays a fundamental role in trust calibration, especially in regulatory and safety-critical domains where interpretability is not negotiable [52].

In particular, the system engineers expressed the greatest appreciation for this transparency and logic structure. They highlighted that the system's consistent reasoning steps and explicit reference to domain-specific keywords made it easier to trace clause applicability. Compared to other participants, they were more willing to accept the recommendations of the tool, provided that the justification was clear, even if the final decision still remained with a human. This suggests that transparency not

only underpins trust, but also aligns well with engineering mindsets that value explainability and traceability. Project managers, on the contrary, tended to prioritise efficiency over insight. And preferred a streamlined experience where they could simply press a button and receive a recommendation, showing less interest in the underlying reasoning.

6.2.2 Theme 2: Time Savings vs. Double Work: A Trade-Off

Theme 2 addressed efficiency expectations. Participants frequently mentioned the time-consuming nature of compliance analysis, especially in the early stages of the project. Several noted that the model could significantly reduce hours spent in cross-functional workshops, aligning with previous findings that early filtering can yield large productivity gains [93]. However, others expressed concern that if the model's outputs must still be fully verified, it could become "double work" rather than a shortcut. In particular, while concerns about output reliability were raised, the execution times of the model itself did not. As shown in Table 6.1, participants considered these run-times negligible compared to the extensive manual effort typically required for the compliance analysis.

At the same time, many participants clarified that accuracy was not the primary concern as long as the system could reliably flag all potentially applicable clauses. As one participant explained, 'I don't mind checking 100 items, as long as I'm not missing number 101.' This suggests that users are less sensitive to false positives than to missed obligations. In this perspective, the system's value is not in perfect classification but in its ability to act as a filter, safely surfacing all ambiguous cases for human review.

Therefore, the 77% accuracy achieved by the best-performing configuration was interpreted not as a failure to automate, but as a signal of how much manual effort could be safely delegated. From this perspective, every clause that can be reviewed confidently and correctly skipped represents net time saved. This aligns with human-in-the-loop deployment principles, where automation acts as a supportive sieve rather than a final judge.

6.2.3 Theme 3: False Negatives as Critical Barriers to Adoption

The most serious technical adoption blocker emerged in theme 3: the risk of false negatives. Participants viewed any missed applicable requirement as a potential compliance or safety failure. This reflects technical trust, where confidence depends on the model's ability to perform reliably under domain-specific constraints. Some participants said they could tolerate false positives ('extra work'), but not false negatives ('project risk'). This aligns with regulatory priorities, where recall is often more important than precision [94]. Even if users trust the organisation, they will not rely on a model that fails to highlight critical items. To maintain trust, future

systems must be conservative in exclusion and generous in detecting ambiguity.

Furthermore, while concern about false negatives is legitimate and deeply rooted in compliance culture, it is also important to contextualise the expectation. No machine learning model, regardless of domain, can guarantee perfect recall. This limitation is not a flaw in the proposed system, but a well established constraint of statistical learning methods in general [95].

Expecting zero false negatives assumes perfect legal interpretation (which even humans disagree on) or a manual fallback on every clause. Both scenarios undermine the productivity gains intended by the automated system. As such, a more realistic view is to use the system as a high-recall filter, accepting that some review remains necessary but reducing the total burden.

Additionally, the presence of a Human-in-the-Loop (HITL) review stage acts as a critical safeguard. This workflow ensures that no decision is final without expert verification. It mirrors common design patterns in safety-critical AI, where machines flag but humans decide [96]. Rather than replacing responsibility, the system augments it, providing traceable, explainable first-pass assessments that help teams focus their attention.

Ultimately, the adoption of AI in regulated engineering environments will require a shift from unrealistic perfection expectations toward risk management, assisted decision-making, a model already accepted in fields such as medical diagnosis [97]. In that sense, this thesis proposes not a replacement for accountability, but a tool to sharpen and support it.

6.2.4 Theme 4: Integration and Feature Needs Are Central to Usability

Theme 4 emphasised that for the prototype to be usable, it must go beyond technical classification. Participants requested further integration with existing tools (e.g., SystemWeaver), exportable summaries for reporting, and smart filtering, especially for functional safety requirements. This indicates that technical performance is necessary but not sufficient. Integration into real workflows is a critical adoption factor. It also confirms broader trends in the design of human-AI interactions, which emphasise the importance of the surrounding infrastructure to allow meaningful use [93]. However, participants were generally positive about the current set of features. Although they saw clear value in integration and some minor interface enhancements, they did not express major unmet needs or request new functionality. In fact, several participants noted that the prototype already provided most of what they would expect from a useful support tool. This suggests that core capabilities, such as semantic matching, clause justification, and structured output, align well with user expectations and may already meet the baseline requirements for practical deployment.

6.2.5 Theme 5: Output Quality Adds Value Beyond Prediction

An unexpected insight in theme 5 was the perceived value of the model-generated explanations. Some participants explicitly stated that the conclusions were clearer than what they would have written themselves. This highlights the potential of the system not just as a predictor, but as a structured documentation assistant. In compliance contexts, where traceability and clarity are critical, this kind of generative value is not trivial. Recent work on language generation for regulated domains confirms this trend, showing that structured summaries can enhance both communication and auditability [94].

However, this added value was less apparent when reviewing outputs from EU directives. Participants noted that while the model's reasoning 'looked good', they felt unqualified to verify its correctness due to the abstract and legalistic nature of the directives. Unlike harmonised standards such as EN 1175, which map closely to engineering practices, directives are often too high-level for clause-by-clause technical interpretation. This highlights a practical limitation of applying our system to legal sources that lack operational detail and underscores the continued importance of harmonised standards for engineering compliance work.

6.2.6 Theme 6: Adaptability and Learning Are Expected Features

Finally, theme 6 revealed that users did not view the model as a fixed tool. Instead, they expected it to improve over time, either by learning from prior inputs or adapting to changing standards. Several participants suggested that if the tool can evolve through feedback, it would earn more trust and reduce future verification needs. This aligns with the concept of augmented intelligence [93], where tools adapt alongside expert workflows. It also suggests the value of building a feedback loop, either through active learning, annotation refinement, or user-adjusted thresholds.

Summary and Implications

The six themes indicate that an AI-assisted pipeline can already reduce compliance workload, but only when certain sociotechnical preconditions are satisfied.

RQ1: Themes 2 and 5 demonstrate that the retrieval + LLM system effectively achieves its main goal. Highlights the most potentially relevant clauses at an initial stage, provides explicit explanations, and condenses numerous pages of regulation into a concise and annotated list of results. Some interviewees even mentioned that the model's text was clearer than their personal notes, providing clear evidence that a well-trained system can generate links that are truly beneficial in routine engineering tasks.

RQ2: Themes 1, 2, and 4 underscore that the adoption of a tool is influenced more by its compatibility with workflows than just by its performance metrics.

Engineers are unlikely to use the tool unless its results are incorporated *within* SystemWeaver, maintain trace-links, and display confidence indicators that enable a predictable review process. Thus, the key elements driving adoption are integration, traceability, and transparent accountability.

RQ3: Themes 1, 3, 5, and 6 focus on trust and explainability. Users may encounter more false positives (which means extra work), but not false negatives (which result in missed obligations). They appreciated the chain-of-thought explanations, referring to them as "living documentation," and anticipate that the system will adapt from their corrections across various projects. This highlights the importance of having calibrated confidence scores and an effective feedback mechanism.

6.3 Limitations & Delimitations

Limitations

This study is subject to a number of limitations which can be grouped into two categories: inherent limitations related to the current state of technology and context, and addressable limitations that can be addressed in future work.

A fundamental limitation lies in the variability and non-determinism of LLMs. These models can generate inconsistent outputs for similar inputs due to their probabilistic nature, especially under non-zero temperature settings. Although prompt engineering and parameter tuning help reduce this effect, full consistency cannot be guaranteed without access to deterministic models. In addition, regulatory clauses often contain ambiguous or context-dependent language, which even domain experts may interpret differently. This introduces subjectivity into both labelling and model evaluation.

Another inherent constraint is the limited size and scope of the ground-truth dataset. The evaluation set consists of 256 clause–project pairs, derived from the EN 1175 standard and a single product line. Although this allows for a tightly scoped evaluation, it limits the generalisability of the results to other domains, products, or regulatory frameworks.

However, several limitations are more remediable. The current system lacks a closed-loop feedback mechanism. Although the prototype supports human-in-the-loop (HITL) validation, it does not yet incorporate that feedback to improve future predictions. Integrating an active learning strategy or retraining pipeline could progressively improve accuracy and adaptability.

Another addressable limitation concerns the domain coverage of the keyword lexicon and embedding models. The prototype is tuned to electrical propulsion systems, and performance may degrade when applied to unrelated domains. However, this can be mitigated by expanding the domain lexicon through expert workshops and fine-tuning embeddings using additional annotated data.

A practical challenge lies in the frequency with which the regulations are updated. Regulatory frameworks, such as EU directives or harmonised standards, can change on an annual basis. The current system assumes static regulatory input and lacks version control or update monitoring mechanisms. As a result, prior applicability classifications may become outdated over time. Addressing this would require developing a semi-automated pipeline to detect new versions of standards, identify modified clauses, and trigger reclassification. A mechanism like this is practical and would significantly improve long-term reliability.

Finally, while the system includes explainability features such as clause similarity scores and LLM-generated reasoning, these may not meet all stakeholder needs. Some users may require structured legal justifications or cross references to internal standards that go beyond the current output format.

Delimitations

To manage data shortage, this study leverages pre-trained language models, unsupervised embedding-based retrieval, and zero-shot reasoning. These techniques reduce dependence on large labelled datasets and enable generalisation to unseen clauses. Although this approach may not match the performance of a fully supervised system trained on domain-specific data, it provides a practical and deployable solution in data-constrained industrial contexts.

To compensate for incomplete project context, domain-relevant keywords were extracted through expert workshops and analysed from high-level Target Description (TD) files. These were integrated into prompt templates and keyword matching algorithms to simulate contextual understanding without requiring full system models.

In terms of scope, the system was developed and evaluated within the context of Volvo Penta's SystemWeaver environment and regulatory focus areas. Broader generalisation would require adaptation to other compliance tools, engineering artefact schemas, and international legal frameworks. In addition, the study focusses on clause-level applicability classification and does not address downstream compliance activities such as testing, certification, or regulatory auditing. These remain important directions for future research.

Expert feedback was collected through iterative prototype demonstrations and structured interviews. This helped refine prompt design, interface layout, and output format, ensuring that the system aligns closely with actual engineering workflows and user expectations.

6.4 Threats to Validity

Internal Validity

In the quantitative evaluation, all test conditions were kept constant except for variations in the embedding model and LLM, ensuring that the observed performance

changes could be attributed to these variables. Ground truth labels were derived from compliance conclusions curated by experienced engineers and domain experts, increasing annotation reliability. Although two annotators labelled the data collaboratively, no formal inter-rater agreement metric was calculated, which limits our ability to quantify annotation consistency. Furthermore, the data set itself, derived from expert conclusions in SystemWeaver, may contain errors, as even domain experts can disagree or make mistakes. Although this introduces some uncertainty, the annotations serve as a practical ground truth, reflecting the current best understanding used in real-world compliance workflows.

To reduce interviewer bias in the qualitative evaluation, all interviews followed a semi-structured format with open questions and minimal prompting. Each participant was individually interviewed to avoid conformity effects or groupthink. However, the interview guide was reviewed only by the thesis supervisor. Best practice recommends that interview protocols be reviewed by multiple experts or tested in a pilot with representative users to assess the clarity of the question and reduce the bias of wording [98]. Additionally, no inter-rater reliability check was applied to the thematic coding of responses, which could affect the consistency of the qualitative interpretation.

External Validity

Although the model design, which includes semantic embeddings, similarity search, and LLM-based classification, is widely useful for regulatory compliance tasks, some elements are still specific to particular domains. The curated keyword lexicon was developed specifically for Volvo Penta's product categories and cannot be generalised to fields such as aerospace, medical devices, or financial systems. Similarly, the prompt templates assume familiarity with engineering concepts and structure drawn from SystemWeaver artefacts, which limits their portability without adaptation.

Despite these constraints, the modular design of the system enables incremental domain adaptation. For instance, new keyword sets and prompt templates can be substituted with minimal code changes, and the architecture supports re-embedding corpora from other regulatory sources.

Results Validity

To prioritise recall in compliance-sensitive scenarios, the F_2 score was selected as the primary evaluation metric. Among the 16 model configurations tested, substantial variance was observed in LLM performance, while embedding models showed relatively smaller differences. However, due to computational and time constraints, statistical significance tests (e.g. t-tests or bootstrap intervals) were not performed, and performance stability was only measured across five repeated runs for the top-performing model.

The qualitative evaluation used thematic coding of the interview responses and included participants from various engineering and regulatory roles. This diversity

enhanced the robustness of the findings. However, the sample size was relatively small and no formal saturation analysis was performed to confirm the completeness of the themes. Lack of follow-up interviews or member verification also limits the confirmability of interpretations.

6.5 Future Work

The current prototype demonstrates the feasibility of combining semantic retrieval, keyword enrichment, and large language models (LLMs) to support compliance interpretation in engineering contexts. However, several promising directions remain for extending its capabilities and improving industrial applicability.

An important future direction involves extending the current two-step filtering process, which combines semantic analysis with a pre-trained LLM, by incorporating models trained on data from the HITL review process. Lightweight, domain-specific classifiers could handle routine clause filtering with greater speed and interpretability. In the longer term, fine-tuning a custom LLM on this growing labelled dataset may improve prediction accuracy in complex cases. These additions would enhance technical capabilities outlined in RQ1 and improve reliability and trust as discussed in RQ3.

Another direction involves expanding the system integration with the internal tools of Volvo Penta. At present, the model runs on the Azure platform and requires manual data extraction from SystemWeaver, with only the outputs being re-integrated via API. Future iterations should aim to embed the model more deeply within SystemWeaver as a native plug-in. This would allow direct retrieval of product metadata, automatic trigger of compliance checks when standards or designs change, and streamlined feedback mechanisms between compliance assessments and systems engineering. Such an integration would further enhance usability and traceability, directly contributing to RQ2 by aligning the system with engineers' real-world workflows.

More empirical validation is also needed to assess the generalisability of the system. So far, the evaluation has focused on a single standard (EN 1175) within a single project. Future work should test the system across various regulatory frameworks, such as ISO 13849 or ISO 10218, as well as in different engineering domains such as marine propulsion or stationary power systems. This would help confirm that the model is adaptable to varying terminology, document structures, and regulatory cultures. Broader evaluation would not only validate performance under RQ3, but would also ensure that the system can be scaled responsibly across new use cases and product lines.

Another key future challenge lies in model governance and traceability. As AI tools are introduced into high-stakes regulatory decision-making, organisations must ensure that their use is aligned with accountability, auditability, and traceability principles. This includes implementing version control for models and prompts, logging

system outputs, and preserving trace links between input text, model reasoning, and human edits. Such measures are essential for building trust in AI-assisted decisions, especially in consideration of emerging standards like ISO/IEC 42001 on AI governance [99]. These governance capabilities are especially relevant for long-term deployment and contribute to the transparency and trust dimensions of RQ2 and RQ3.

Finally, as the system scales, practical concerns such as inference cost and response time will become increasingly important. Future work could investigate hybrid architectures that combine fast, shallow models with LLMs, apply prompt optimisation to reduce token use, or evaluate the deployment of compressed models for less critical tasks. These optimisations would enhance the system's sustainability and responsiveness without compromising its interpretability or reliability.

7

Conclusion

This thesis has demonstrated the potential of combining representation learning techniques with large language models (LLMs) to support regulatory compliance in engineering domains. Addressing the first research question (RQ1), a two-phase pipeline was developed to map regulatory clauses and engineering artifacts into a shared semantic space, followed by clause-level classification using an LLM. Evaluated against an expert-annotated dataset, the system achieved 77.5% accuracy and 94.3% recall, resulting in an F_2 score of 88.9%. These results show that the method is particularly effective for high-recall filtering, making it well suited for compliance checks where completeness is critical.

The second research question (RQ2) examined integration, usability, and transparency. The prototype was linked to SystemWeaver and supports automatic upload of results with traceability to relevant engineering artefacts. To improve transparency, features such as confidence scores and editable justifications were added. The engineers appreciated being able to review the reasoning of the model and saw the potential of the tool to support gate reviews, reduce preparation time, and improve the traceability of documentation. However, they stressed that the ultimate responsibility should remain with human experts.

For the third research question (RQ3), both performance metrics and user feedback indicated that the system's outputs were generally clearer and more structured than previous documentation. Experts were willing to accept some false positives to achieve high recall, but pointed out that false negatives could be more problematic. This underlined the importance of keeping a human-in-the-loop to review and validate results.

There are some limitations to the study. The focus was primarily on the EN 1175 standard, which may restrict the extent to which the approach can be applied. Additionally, the LLM occasionally produced overly verbose or inaccurate content. Future work should explore active-learning loops driven by user corrections, automatic rechecks when artefacts or regulations change, broader validation across additional standards and product lines, and the introduction of governance tooling. Such as audit logs, model cards, and rollback mechanisms, to meet emerging AI-assurance requirements.

Overall, the findings serve as an initial validation of the potential for integrating semantic embeddings, large language models, and human expertise to improve the efficiency and interpretability of compliance analysis. Although the system does not replace expert judgment, it helps prioritise areas for review, suggesting a promising direction for AI-augmented compliance engineering in Volvo Penta and similar contexts. However, further research and iterative development are required to fully assess scalability, reliability, and domain adaptation.

Bibliography

[1] Compliance & Risks. How to do regulatory impact interpretation on your products, 2025. Accessed: March 20, 2025.

[2] IntegrityRisk International. Eu regulatory compliance trends 2022, 2022. Accessed: March 20, 2025.

[3] John Slankas and Laurie Williams. Automated extraction of non-functional requirements in available documentation. In *2013 1st International workshop on natural language analysis in software engineering (NaturaLiSE)*, pages 9–16. IEEE, 2013.

[4] European Commission. Restriction of hazardous substances (rohs) and reach. `https://environment.ec.europa.eu/topics/waste-and-recycling/rohs-directive_en`, 2023. Accessed May 2025.

[5] European Union. Electromagnetic compatibility (emc) directive 2014/30/eu. `https://en.wikipedia.org/wiki/Electromagnetic_compatibility`, 2023. Accessed May 2025.

[6] United Nations Economic Commission for Europe. Cybersecurity and software update regulations for automotive vehicles (un r155 and un r156). `https://arxiv.org/abs/2407.00483`, 2024. Accessed May 2025.

[7] Miroslaw Staron, Wilhelm Meding, and Jonas Höglund. Challenges of compliance in embedded systems engineering. *Journal of Systems and Software*, 174:110887, 2021.

[8] Yifan Zhang, Ravi Kumar, and Jan Bosch. Leveraging nlp techniques for regulatory document analysis in safety-critical domains. In *Proceedings of the 30th IEEE International Requirements Engineering Conference (RE)*, pages 125–135. IEEE, 2022.

[9] Organisation for Economic Co-operation and Development (OECD). Regulatory policy outlook 2020, 2020. Accessed: 2025-05-23.

[10] International Electrotechnical Commission (IEC). White paper: Smart manufacturing and the role of standards in regulatory compliance, 2021. Accessed:

2025-05-23.

[11] Jiansong Zhang and Nora M El-Gohary. Semantic nlp-based information extraction from construction regulatory documents for automated compliance checking. *Journal of computing in civil engineering*, 30(2):04015014, 2016.

[12] Amin Sleimi, Nicolas Sannier, Mehrdad Sabetzadeh, Lionel Briand, and John Dann. Automated extraction of semantic legal metadata using natural language processing. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pages 124–135. IEEE, 2018.

[13] Olivier Amaral, Sana Abualhaija, and Lionel C. Briand. Ml-based compliance verification of data processing agreements against gdpr. In *Proc. 31st IEEE International Requirements Engineering Conference (RE)*, pages 90–101, 2023.

[14] Riad Sonbol, Ghaida Rebdawi, and Nada Ghneim. The use of nlp-based text representation techniques to support requirement engineering tasks: A systematic mapping review. *arXiv preprint arXiv:2206.00421*, 2022.

[15] European Union. Types of legislation. `https://european-union.europa.eu/institutions-law-budget/law/types-legislation_en`, n.d. `https://european-union.europa.eu/institutions-law-budget/law/types-legislation_en`.

[16] Iso – international organization for standardization. `https://www.iso.org/about-us.html`, 2025. Accessed: 2025-05-19.

[17] Iec – international electrotechnical commission. `https://www.iec.ch/about`, 2025. Accessed: 2025-05-19.

[18] Imo – international maritime organization. `https://www.imo.org/en/About/Pages/Default.aspx`, 2025. Accessed: 2025-05-19.

[19] International Organization for Standardization. Iso 26262: Road vehicles – functional safety, 2018. ISO 26262-1:2018.

[20] CEN and CENELEC. European standards. `https://www.cencenelec.eu/european-standardization/european-standards/`, n.d. `https://www.cencenelec.eu/european-standardization/european-standards/`.

[21] Din – deutsches institut für normung e.v. `https://www.iso.org/member/1511.html`, 2025. Accessed: 2025-05-19.

[22] Bsi – national standards body. `https://www.bsigroup.com/en-US/about-bsi/national-standards-body/`, 2025. Accessed: 2025-05-19.

[23] Sis – swedish institute for standards. `https://www.iso.org/member/2101.html`, 2025. Accessed: 2025-05-19.

[24] European Commission. Harmonised standards. https://single-market-economy.ec.europa.eu/single-market/european-standards/harmonised-standards_en, n.d. https://single-market-economy.ec.europa.eu/single-market/european-standards/harmonised-standards_en.

[25] BSI Group. Bs en 1175:2020 - safety of industrial trucks - electrical/electronic requirements. https://knowledge.bsigroup.com/products/safety-of-industrial-trucks-electrical-electronic-requirements/standard, 2020. https://knowledge.bsigroup.com/products/safety-of-industrial-trucks-electrical-electronic-requirements/standard.

[26] SAE International. Sae j1939 - the vehicle bus standard. https://www.sae.org/standards/content/j1939/, n.d. https://www.sae.org/standards/content/j1939/.

[27] AUTOSAR. Autosar classic platform specifications. https://www.autosar.org/standards/classic-platform/, n.d. https://www.autosar.org/standards/classic-platform/.

[28] Volvo Group. Corporate standards. https://www.volvogroup.com/en/suppliers/useful-links-and-documents/corporate-standards.html, n.d. https://www.volvogroup.com/en/suppliers/useful-links-and-documents/corporate-standards.html.

[29] Bashar Nuseibeh and Steve Easterbrook. Requirements engineering: A roadmap. *Proceedings of the Conference on the Future of Software Engineering*, pages 35–46, 2000.

[30] Robert G. Cooper. *Winning at New Products: Creating Value Through Innovation*. Basic Books, 4th edition, 2019.

[31] Tony Gorschek and Claes Wohlin. A model for technology transfer in practice. *IEEE Software*, 23(6):88–95, 2006.

[32] Christof Ebert and Jan De Man. Requirements engineering: A good practice guide. *IEEE Software*, 33(5):81–85, 2016.

[33] SystemWeaver AB. Systemweaver, 2024. Accessed: 2025-05-23.

[34] Systemite AB. Systemweaver documentation. https://support.systemweaver.se. Accessed: 2025-05-15.

[35] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint*, 1301.3781, 2013.

[36] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[38] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, pages 4171–4186, 2019.

[39] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *EMNLP-IJCNLP*, pages 3982–3992, 2019.

[40] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, and Chris Tar. Universal sentence encoder. *EMNLP Workshop on Representation Learning for NLP*, pages 169–174, 2018.

[41] Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. In *NeurIPS*, pages 7059–7069, 2019.

[42] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Legal-bert: The muppets straight out of law school. In *Findings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP Findings)*, pages 2898–2904. Association for Computational Linguistics, 2020.

[43] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

[44] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33*, pages 1877–1901. Curran Associates, Inc., 2020.

[45] Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735*, 2023.

[46] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[47] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi,

Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[48] OpenAI Community. Cheat sheet: Mastering temperature and top-p in chatgpt api, 2023. Accessed: 2025-05-06.

[49] Orlena C Z Gotel and Anthony CW Finkelstein. An analysis of the requirements traceability problem. pages 94–101, 1994.

[50] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann, 1994.

[51] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[52] Zachary C. Lipton. The mythos of model interpretability. In *Communications of the ACM*, volume 61, pages 36–43, 2018.

[53] Alejandro Barredo Arrieta et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. In *Information Fusion*, volume 58, pages 82–115. Elsevier, 2020.

[54] Leilani Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89. IEEE, 2018.

[55] 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 *Conceptual Modeling – ER 2008*, volume 5231 of *Lecture Notes in Computer Science*, pages 154–168. Springer, 2008.

[56] Travis D. Breaux and Annie I. Antón. Analyzing regulatory rules for privacy and security requirements. In *IEEE Transactions on Software Engineering*, volume 34, pages 5–20, 2008.

[57] Nolan Zhang, Peter Bodorik, and Dawn N. Jutla. Compliance of privacy policies with canadian PIPEDA. In *Proceedings of the International Conference on e-Business (ICE-B)*, pages 277–284, 2007.

[58] Fahad ul Hassan and Tuyen Le. Automated requirements identification from construction contract documents using natural language processing. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, 12(2):04520009, 2020.

[59] Dareen M. Salama and Nora M. El-Gohary. Semantic text classification for supporting automated compliance checking in construction. *Journal of Computing*

in Civil Engineering, 30(1):04014106, 2016.

[60] Peng Zhou and Nora M. El-Gohary. Ontology-based multi-label text classification for enhanced automatic compliance checking. *Journal of Computing in Civil Engineering*, 30(5):04015057, 2016.

[61] Amin Sleimi, Nicolas Sannier, Mehrdad Sabetzadeh, Lionel Briand, Marcello Ceci, and John Dann. An automated framework for the extraction of semantic legal metadata from legal texts. *Empirical Software Engineering*, 26:1–50, 2021.

[62] Olivier Amaral, Sana Abualhaija, Daniele Di Pompeo, Mojtaba Sabetzadeh, and Lionel C. Briand. Ai-enabled automation for completeness checking of privacy policies. *IEEE Transactions on Software Engineering*, 48(11):4212–4229, 2022.

[63] Ilias Chalkidis, Emmanouil Fergadiotis, Nikolaos Manginas, Eva Katakalou, and Prodromos Malakasiotis. Regulatory compliance through doc2doc information retrieval: A case study in eu/uk legislation where text similarity has limitations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 3498–3511, 2021.

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

[65] Bing Wen, Tingjun Wang, Jiawei Xu, Ying Liu, Jinxun Li, and Shuhong Lin. File compliance detection using a word2vec-based semantic similarity framework. *Informatica*, 49(18):51–66, 2025.

[66] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. How does nlp benefit legal system: A summary of legal artificial intelligence. *arXiv preprint arXiv:2004.12158*, 2020.

[67] Seonghyeon Moon, Seokho Chi, and Seok-Been Im. Automated detection of contractual risk clauses in construction specifications using bert. *Automation in Construction*, 142:104465, 2022.

[68] Daniel Braun and Florian Matthes. Clause topic classification in german and english standard-form contracts. In *Proceedings of the 5th Workshop on e-Commerce and NLP (ECNLP 5) at ACL 2022*, pages 199–209. Association for Computational Linguistics, 2022.

[69] Venkatraman Balasubramanian. Large language models: Extracting and summarizing regulatory intelligence from health authority guidance documents. *DIA Global Forum*, 2024(Jan):16–19, 2024.

[70] Shabnam Hassani, Mehrdad Sabetzadeh, Daniel Amyot, and Jain Liao. Rethinking legal compliance automation: Opportunities with large language mod-

els. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 432–440. IEEE, 2024.

[71] Shuyi Wu, Jiajie Liang, Hao Chen, and Yueting Zhao. Reg-llama: Domain-specific language models for regulatory compliance. In *Proceedings of the ACL Workshop on Financial Technology and NLP (FinNLP 2025)*, pages 120–131, 2025.

[72] Zhiyu Cao and Zachary Feinstein. Large language model in financial regulatory interpretation. *arXiv preprint arXiv:2405.06808*, 2024.

[73] RadarFirst. Human in the loop is essential for ai-driven compliance. `https://www.radarfirst.com/blog/why-a-human-in-the-loop-is-essential-for-ai-driven-privacy-compliance/`, 2024. Accessed: 2025-05-08.

[74] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164, 2009.

[75] Klaas-Jan Stol and Brian Fitzgerald. The abc of software engineering research. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 27(3):1–51, 2018.

[76] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

[77] OpenAI. New and improved embedding model, 2023. Accessed: 2025-05-18.

[78] Databricks. Gte models on databricks marketplace, 2024. Accessed: 2025-05-18.

[79] Berk Atil, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, Zhe Wu, Lixinyu Xu, and Breck Baldwin. Non-determinism of "deterministic" llm settings. *arXiv preprint arXiv:2408.04667*, 2024.

[80] OpenAI. Gpt-4o mini: Advancing cost-efficient intelligence. `https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/`, 2024. Accessed: 2025-05-08.

[81] OpenAI. Gpt-3.5 turbo model documentation. `https://platform.openai.com/docs/models/gpt-3.5-turbo`, 2023. Accessed: 2025-05-08.

[82] Meta AI. Introducing meta llama 3: The most capable openly available llm to date. `https://ai.meta.com/blog/meta-llama-3/`, 2024. Accessed: 2025-05-08.

[83] Anthropic. Claude 3.7 sonnet. `https://www.anthropic.com/claude/sonnet`, 2025. Accessed: 2025-05-08.

[84] Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. The good, the bad, and the greedy: Evaluation of llms should not ignore non-determinism. *arXiv preprint arXiv:2407.10457*, 2024.

[85] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[86] Carolyn B. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557–572, 1999.

[87] Michael Quinn Patton. *Qualitative Research & Evaluation Methods: Integrating Theory and Practice*. Sage Publications, Thousand Oaks, CA, 4th edition, 2015.

[88] Jakob Nielsen and Thomas K. Landauer. A mathematical model of the finding of usability problems. In *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, pages 206–213, Amsterdam, The Netherlands, 1993. ACM.

[89] Greg Guest, Arwen Bunce, and Laura Johnson. How many interviews are enough? an experiment with data saturation and variability. *Field Methods*, 18(1):59–82, 2006.

[90] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, 2006.

[91] Daniela S. Cruzes and Tore Dybå. Recommended steps for thematic synthesis in software engineering. In *Proceedings of the International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 275–284. IEEE, 2011.

[92] Harsimran Kaur, Richard Zhang, Jeffrey L. Huang, Saleema Amershi, Ece Kamar, Daniel S. Weld, Besmira Nushi, John Lee, Eric Horvitz, and Jaime Teevan. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2020.

[93] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kristina Holtz, and Eric Horvitz. Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[94] Yu Zhang, Daniel Preoţiuc-Pietro, Muthu Chidambaram, and Yang Liu. Towards explainable and trustworthy ai for compliance document generation. *Pro-*

ceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 819–834, 2021.

[95] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

[96] Andreas Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, 2016.

[97] Mark P Sendak, Jason D'Arcy, Sravan Kashyap, Michael Gao, Mark Nichols, Kenney Corey, William Ratliff, and Suresh Balu. "the human body is a black box": Supporting clinical decision-making with deep learning. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

[98] Milagros Castillo-Montoya. Preparing for interview research: The interview protocol refinement (ipr) framework. *The Qualitative Report*, 21(5):811–831, 2016.

[99] International Organization for Standardization. Iso/iec 42001:2023 - artificial intelligence management system, 2023. Available at: `https://www.iso.org/standard/81228.html`.

A

TD-Cited Standards

Table A.1: Standards formally cited in the TD for the anonymised electric propulsion project.

Standard or Directive	Application Area	Included in Study?
Low Voltage Directive 2006/95/EG	General Electrical Safety	Yes
EMC Directive 2014/30/EC	Electromagnetic Compatibility	Yes
Directive 2006/66/EC (Batteries)	Hazardous Substances in Batteries	Yes
EU 2018/858	Vehicle Type Approval (On-Road)	No
ECE R10	EMC for Vehicle Components	No
ECE R100	Battery Safety (Vehicles)	No
US GHG / CARB ZEP	US Market Emissions	No
Machine Directive 2006/42/EC	Machinery Safety	**Partially** (via EN 1175)
Outdoor Noise Directive 2000/14/EC	Environmental Noise	No
Directive 2012/19/EU (WEEE)	Waste Electrical & Electronic Equipment	No
Directive 2011/65/EU (ROHS2)	Restriction of Hazardous Substances	No
ISO 14990-1	Safety of Earth-Moving Machinery (Part 1)	No
ISO 14990-3	Safety of Earth-Moving Machinery (Part 3)	No
EN 1175:2020	Electrical Safety for Industrial Trucks	Yes

B

Lexicon of Domain-Specific Terms

The following table lists the 226 unique domain-specific terms used in the semantic matching stage of our analysis, organized by 14 technical categories.

Category	**Terms**
Charging	charging cable, charging inlet, charging connector, charging plug, voltage class B, low voltage, stand still, immobilize, contactors, over current, over charging, short circuit
DCDC	DC/DC converter, converter, power supply, extra low voltage, voltage class B, voltage class A, DC voltage supply, over current, over charging, short circuit
EMC	Radiated emission, RE, Conductive emission, CE, Radiated immunity, RI, Conductive immunity, CI, ESD, magnetic field, EMC
ESS HW (Energy Storage System – Hardware)	Contactors, fuses, pack, cell, module, RESS, EES, cell chemistry, Lithium, li-ion, NMC, Lithium nickel manganese cobalt oxides, LFP, Lithium iron phosphate, anode, cathode, electrolyte, cylindrical, pouch, prismatic, pre charge resistance, busbars, internal resistance, insulation monitoring, insulation, isolation, interlock loop, HVIL, OCV
ESS SW (Energy Storage System – Software)	BMU, battery management unit, BMS, battery management system, SOC, State of charge, SOH, State of health, SOE, State of energy, SOP, state of power, EOL, end of life, SOX, BOL, beginning of life, power limits, current limits, temperature limits, ageing, aging, balancing

Continued on next page

Table B.1 – continued from previous page

Category	Terms
Electric machine	electric machine, electric motor, motor, generator, regenerator, Speed, torque, over speed, shaft, position control, rotor, stator, spin direction, direction of rotation, winding temperature, bearing, bearing current, thermal hotspot, insulation class, demagnetization, torque ripple, balancing grade, IEC 60334, permanent magnets, stall torque, locked rotor, rpm, power factor, Nm, inertia, rotor inertia, peak torque, constant torque, continues torque, rare earth material, frame size, shaft height, drive end, non drive end, radial, axial, flux, radial force, radial flux, axial flux, axial force, interturn short circuit, inter-turn short circuit, neutral point, star point
Electric performance	Withstand voltage, Over voltage category, OVC, Pollution degree, IPXX, IP code, ingress protection code, dielectric, Protective earth, PE, potential equalization point, PEQ, impulse voltage, material group, creapage, clearance, parasitic capacitance, temperature rise, coolant leakage, leakage test, power module
Insulation measurement	insulation, isolation, Ω/V, insulation coordination, clearance distance, creepage distance, frame fault, short circuit
Inverter	electric motor drive, electric machine drive, voltage pulse rise time, power factor correction, active short circuit, short circuit, THD, power factor, DC bus discharge, DC link discharge, DC link capacitance discharge, power module
MDS (Main Drive System)	electric drive systems, electric drive train, propulsion, unintended acceleration, unintended deceleration, traction, propel, propulsion, overvoltage, reflection voltage, voltage reflection, two phase short circuit, three phase short circuit, phase order, Y-connection, delta connection, star connection, triangle connection
OnBC	OnBC, On-board charger, AC/DC converter, converter, Grid supply, power source, power supply, Power factor, Supply voltage, mains, mains supply, AC voltage, voltage class B, Low voltage, three phase, neutral, split phase, single phase, over charging, short circuit
Performance	Power, energy through put, capacity, voltage, peak current, self discharge, battery diagnostics, cell diagnostics

Continued on next page

Table B.1 – continued from previous page

Category	Terms
Safety	Thermal event, TRD, thermal runaway detection, gassing, outgassing, PPR, passive propagation, propagation, cell venting, pack venting, CID, current interrupting device, over charge, over charging, charging, discharging, over voltage, under voltage, over current, short circuit, Over temperature, coolant leakage
ePTO (Electric Power Take-Off)	voltage class B, AC voltage supply, DC voltage supply, Power supply, over current, three phase, neutral, single phase, short circuit

C

Semantic Matching Results

C.1 Domain Keyword Matching

Table C.1: Single-word and complete-phrase keyword hits for the binding parts of the four directives analysed in this thesis.

	EN 1175	Low Voltage Directive	EMC Directive	Battery Directive
Single word matched count	197	28	42	30
Complete words matched count	142	14	12	10

Table C.1 expands the headline figures reported in Section 4.4.2. Counts are derived from two sources: (1) 400 unique single words, obtained through tokenization, lemmatization, stop-word removal, and deduplication from the original 253-term lexicon; and (2) 226 complete unique terms retained after deduplication from the original one. Because the lexicon itself is fixed, tokenisation and lemmatisation are deterministic; hence the numbers do not vary across model configurations used later in the pipeline.

C.2 Product-Section Match Counts

Table C.2: Number of product sections retrieved by each embedding model for the EN 1175 standard and three EU directives. "Binding" indicates clauses that fall inside a document's normative scope.

	EN 1175	Low-Voltage Dir.	EMC Directive	Battery Directive
Legal-BERT	144 (131 binding)	44 (16 binding)	57 (16 binding)	42 (19 binding)
BGE-m3	187 (165)	60 (24)	76 (31)	55 (25)
text-embedding-ada-002	197 (176)	67 (28)	92 (34)	66 (31)
databricks-gte-large-en	207 (186)	66 (25)	100 (38)	61 (27)

Counts are based on the retrieval settings described in 4.4.3 ($k = 10$, $\sigma \geq 0.5$).

C.3 Certainty-Label Statistics

Table C.3: Distribution of certainty labels for each embedding model across the EN 1175 standard and three EU directives. NM: No Match, VCNR: Very Certain Not Relevant, VCR: Very Certain Relevant, PR: Possibly Relevant.

Model	Label	Document			
		EN 1175	Low-Voltage	EMC	Battery
Legal-BERT	NM	40	18	28	15
	VCNR	10	3	4	2
	VCR	29	9	12	9
	PR	177	46	71	42
BGE-m3	NM	29	11	23	8
	VCNR	10	9	9	4
	VCR	38	12	15	11
	PR	179	44	68	45
text-embedding-ada-002	NM	19	7	9	2
	VCNR	15	9	17	9
	VCR	40	14	19	13
	PR	182	46	70	44
databricks-gte-large-en	NM	18	5	5	4
	VCNR	14	10	11	6
	VCR	42	13	20	12
	PR	182	48	79	46

Table C.4: 20^{th}- and 80^{th}-percentile z-score cut-offs (within-document) that define the certainty bands.

Model	Document	z_{20} / z_{80}			
		EN 1175	Low-Voltage	EMC	Battery
Legal-BERT		141 / 29	41 / 9	70 / 12	35 / 9
BGE-m3		107 / 38	28 / 12	55 / 16	24 / 11
text-embedding-ada-002		99 / 40	23 / 14	42 / 19	16 / 14
databricks-gte-large-en		91 / 42	24 / 14	35 / 20	20 / 13

The cut-offs are computed from the non-zero similarity scores described in 4.4.4.

D

LLM Prompt and Inference Code

This appendix presents the complete Python code used to construct the system and user prompts for LLM-based clause classification. The logic follows the methodology described in Section 4.5.2.

```
# Define system prompt
SYSTEM_MESSAGE = (
    "You are a legal/regulatory compliance advisor assisting
       with product requirements in a company called Penta.\n
       "
    "You will receive three information from user: a section
       from a standard or regulation, a high-level project
       description most relevant to this standard or
       regulation section extracted by FAISS, and engineering
        keywords matching this standard or regulation section
       . This engineering keyword list is a list of domain
       terms related to this project. If the project
       description or matched keywords fields are empty, it
       means that there is no project description or keywords
        matched."
    "You need to use the user-provided information to perform
        a core task: determine whether the term regulation is
        a requirement applicable to this project in Penta.
       You need to perform this task in the following three
       steps and output your answer exactly according in the
       following format.\n"

    "1, Choose one of the three answers: Not a Requirement;
       Requirement Not Applicable; Requirement Applicable. To
        answer this, you should firstly decide if this
       standard or regulation section is a requirement or not
       . One hint is that if it uses prescriptive language ('
       shall,' 'must,' 'requires,' 'obligates') or imposes a
       constraint or condition, it is a requirement.
       Otherwisethis section is not a requirement, then
```

```
        choose 'Not a Requirement' for these cases. If you
        think this section is a requirement, you need to
        determine whether this requirement applicable to this
        project.  You should perform this reasoning according
        to the user-provided information which indicates
        relevance between the standard section and product
        description, matched engineering keywords.
        Additionally, during this reasoning process, you
        should also refer those information: Firstly, Penta is
         a supplier of power solutions for marine and
        industrial applications. Therefore, if the subject of
        the regulation or standard section is other components
        , such as steer and brake parts, they are not within
        the scope of application of Penta. Secondly, If the
        content of the clause expresses the requirements that
        the OEM needs to implement (For example, truck
        manufacture is OEM), this clause also does not apply
        to Penta. For these not applicable cases, choose '
        Requirement Not Applicable'. Use all the above
        information to make your assessment. If you think this
         standard or regulation section is relevant to the
        project, please choose 'Requirement Applicable'.\n"

    "2, Output the reasoning process that led you to the
        conclusion in the first step. In other words, please
        output why you judge it to be a requirement and why
        you judge whether it is applicable to this project.\n"

    "3, If it is 'Not a Requirement' or 'Requirement Not
        Applicable' in first step, please briefly summarize
        the content of this clause. If this clause is '
        Requirement Applicable' in first step, please first
        list what the subject is that needs to implement this
        clause. For example, the clause: Connectors for energy
         sources shall conform to Annex A. The subject of this
         clause is Connectors. And then, please output the
        specific requirements. If the clause involves certain
        parameter ranges, such as the noise range should be
        between 75dB and 120dB, please list them separately.
        List all references such as other terms, other
        standards, tables etc.as well as the object that needs
         to meet these references which have been mentioned in
         this clause. Only list those detailed information
        when this clause is Requirement Applicable in first
        step, otherwise just briefly summarize the content of
        this clause\n"

    "Here is additional product application data for
        reference:\n"
```

```
    f"{application_data_string}\n\n"
    "Keep this context in mind when evaluating each match.\n"
)

def build_user_text(row):
    return f"""
Standard or regulation section:
{row['section_title']}
{row['standard_text']}

Project description chunk(s):
{row['product_section_text']}

Matched engineering keywords:
{row["Matched_Domain_Words_complete"]}
"""

def process_row(row):
    """
    Skip calling the LLM if is_binding = False.
    Otherwise, build a prompt and call the LLM.
    """

    if not row.get("is_binding", True):
        return (
            "1. Not a Requirement\n"
            "2. The standard is not binding for this product
               or project.\n"
            "3. No further compliance requirements because it
               's not binding."
        )
    else:
        user_text = build_user_text(row)
        return generate_response_meta(
            SYSTEM_MESSAGE,
            user_text,
            client_meta
        )
```

Listing D.1: Prompt construction and LLM inference code

E

Additional Confusion Matrices

This appendix contains the remaining confusion matrices from the five validation runs performed during model evaluation. As discussed in the main body of the thesis, only one representative matrix was included for clarity, while the rest are shown here for completeness and transparency.

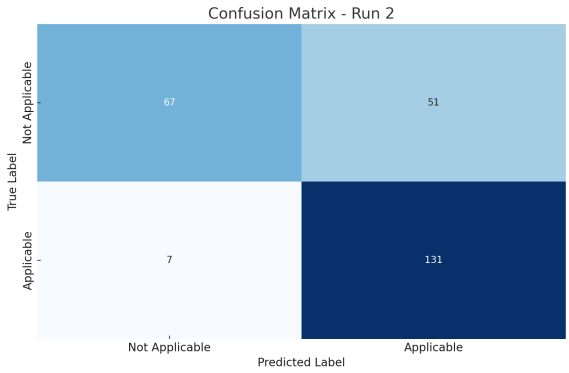

Figure E.1: Confusion matrix for Validation Set (Test 2).

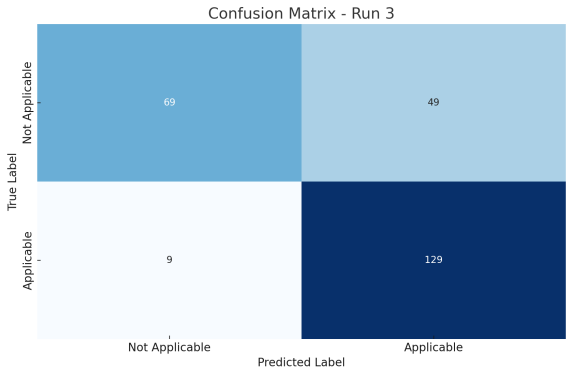

Figure E.2: Confusion matrix for Validation Set (Test 3).

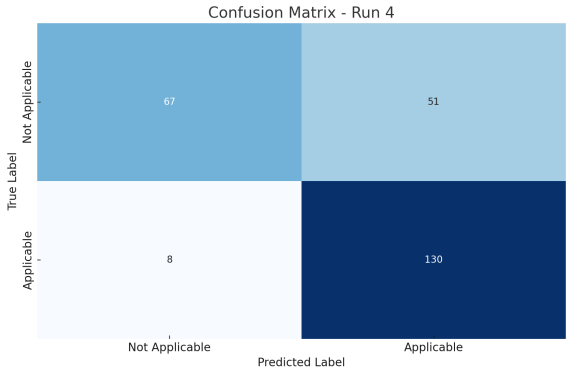

Figure E.3: Confusion matrix for Validation Set (Test 4).

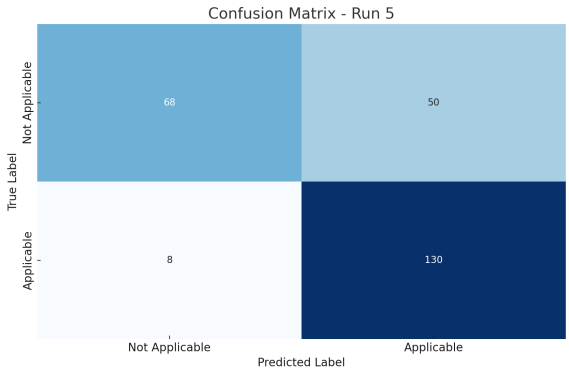

Figure E.4: Confusion matrix for Validation Set (Test 5).

F

User Interface Screenshots

This appendix provides a full walkthrough of the seven-page Streamlit-based prototype, showing the user interface at each major step of the compliance pipeline. Each figure represents either a stage in the pipeline or a specific interaction used during evaluation. The screenshots are taken when using gpt 3.5 turbo and LegalBert.

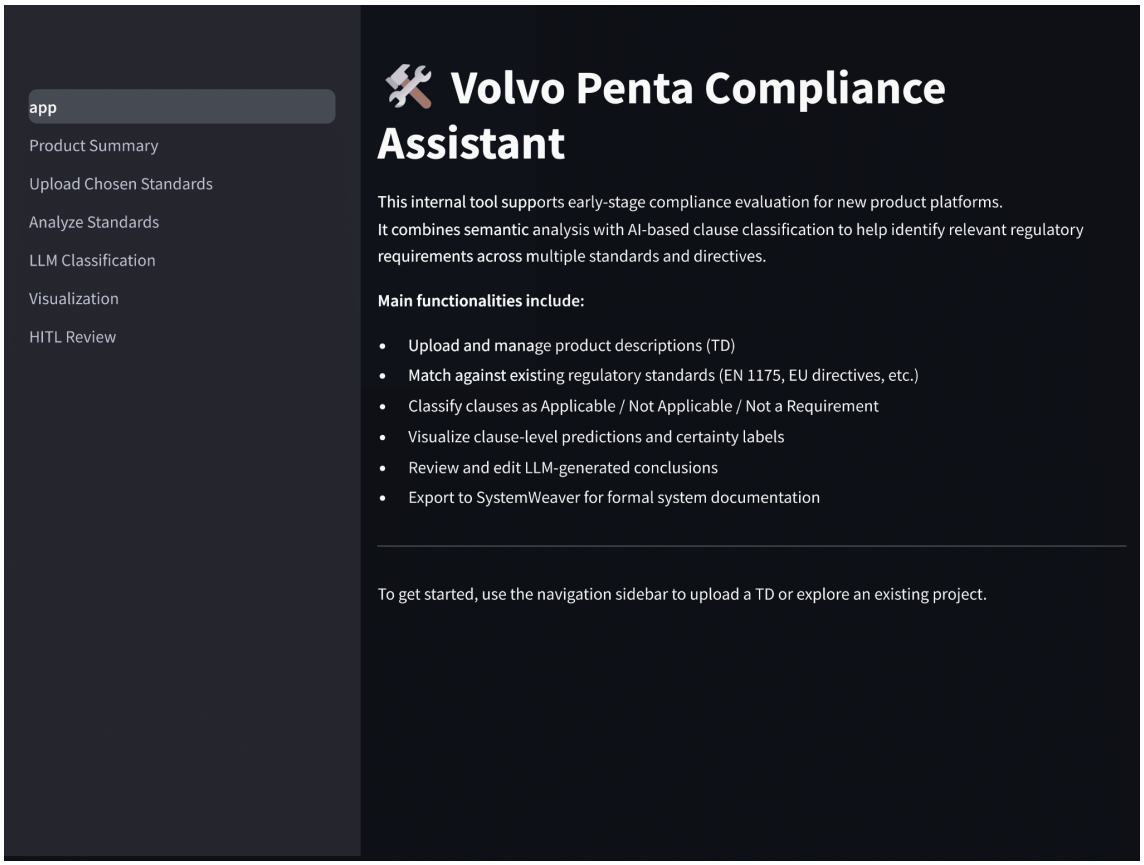

Figure F.1: Welcome page of the prototype. Users are introduced to the system and guided to begin.

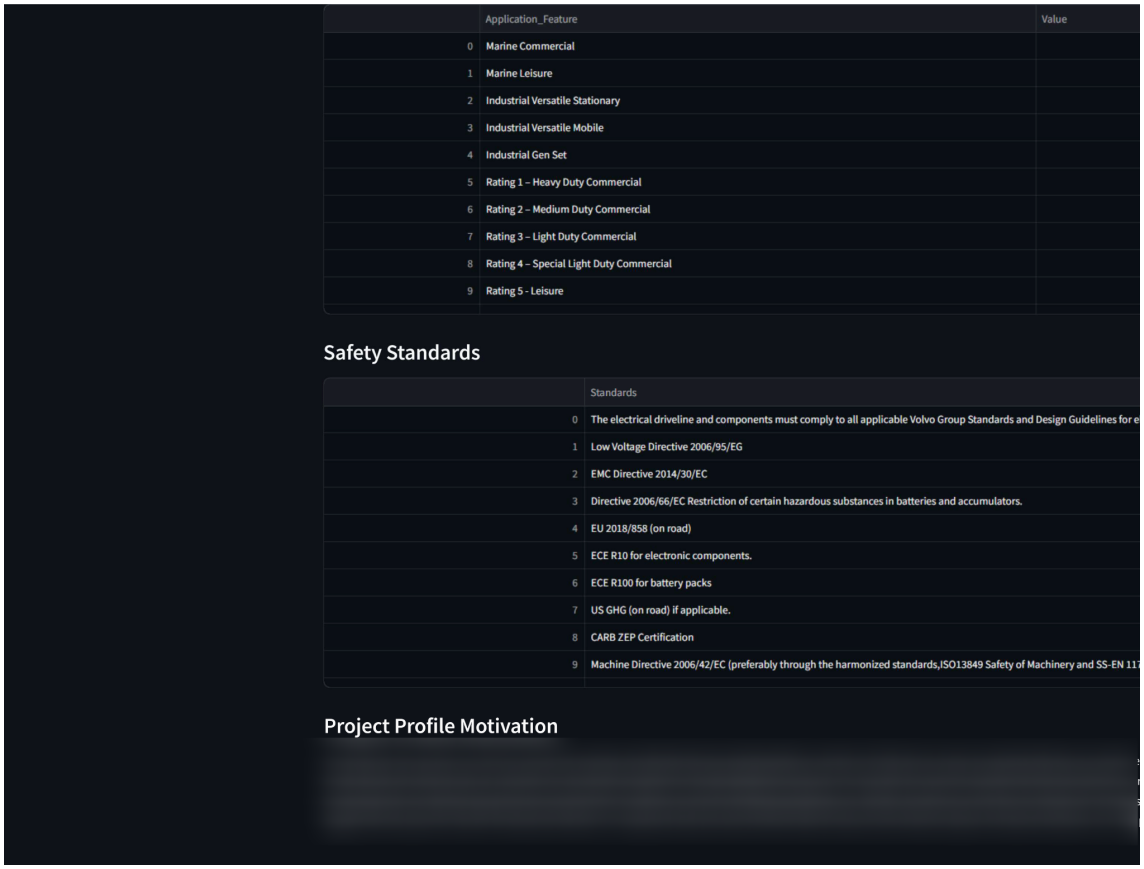

Figure F.2: Page 1 – Product parsing step (top of page).

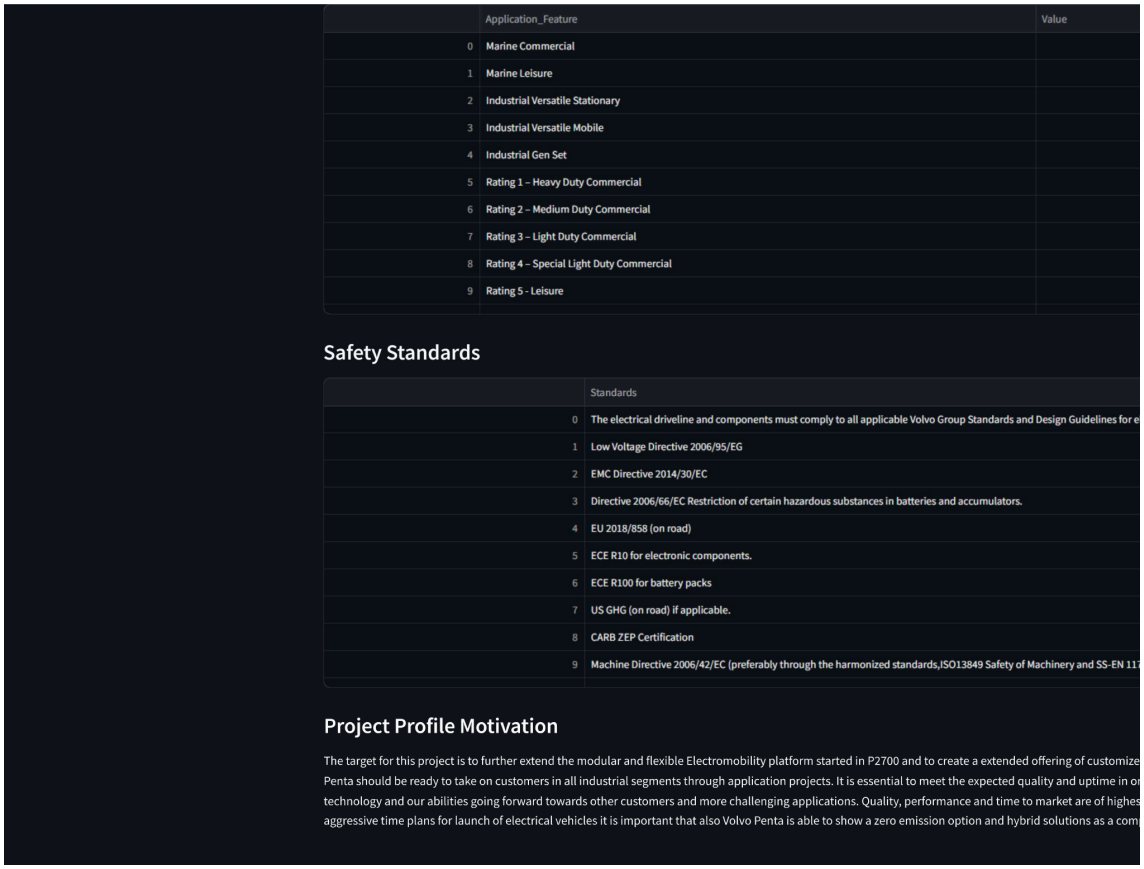

Figure F.3: Page 1 – Domain fields populated (middle of page).

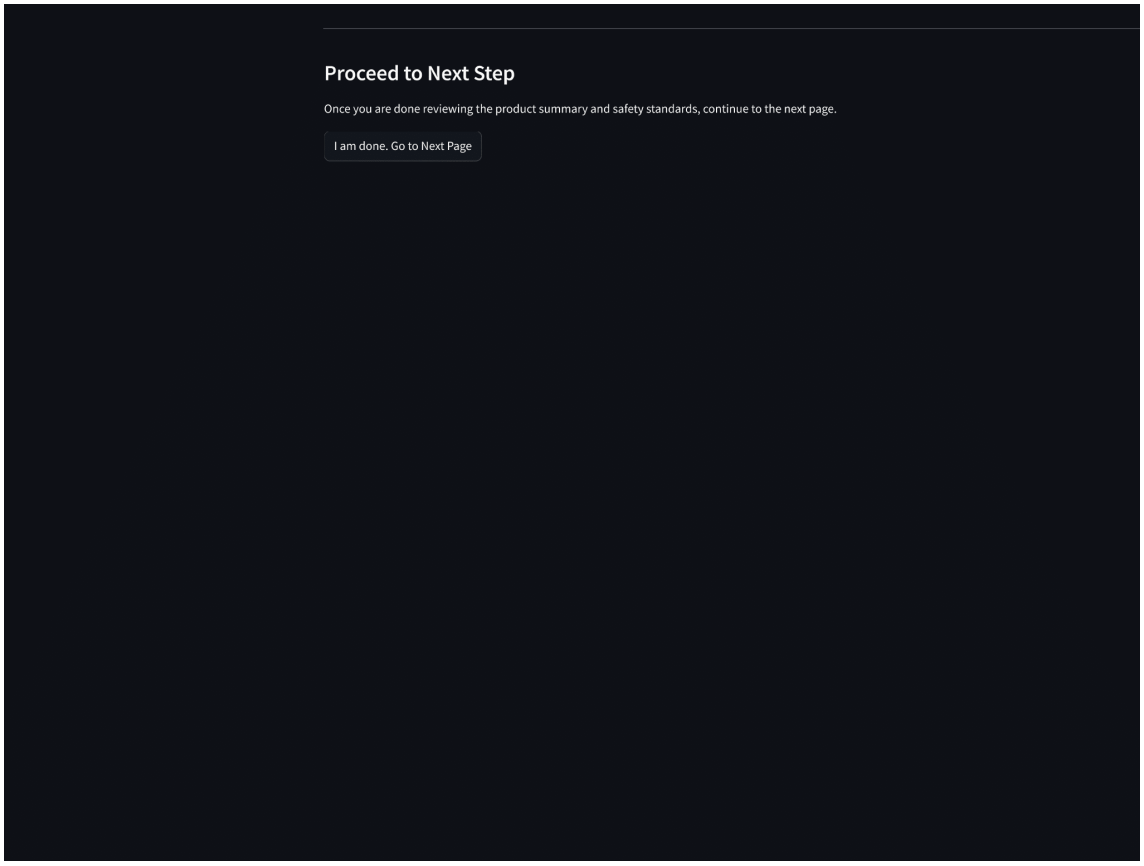

Figure F.4: Page 1 – Proceed to next step button (bottom of page).

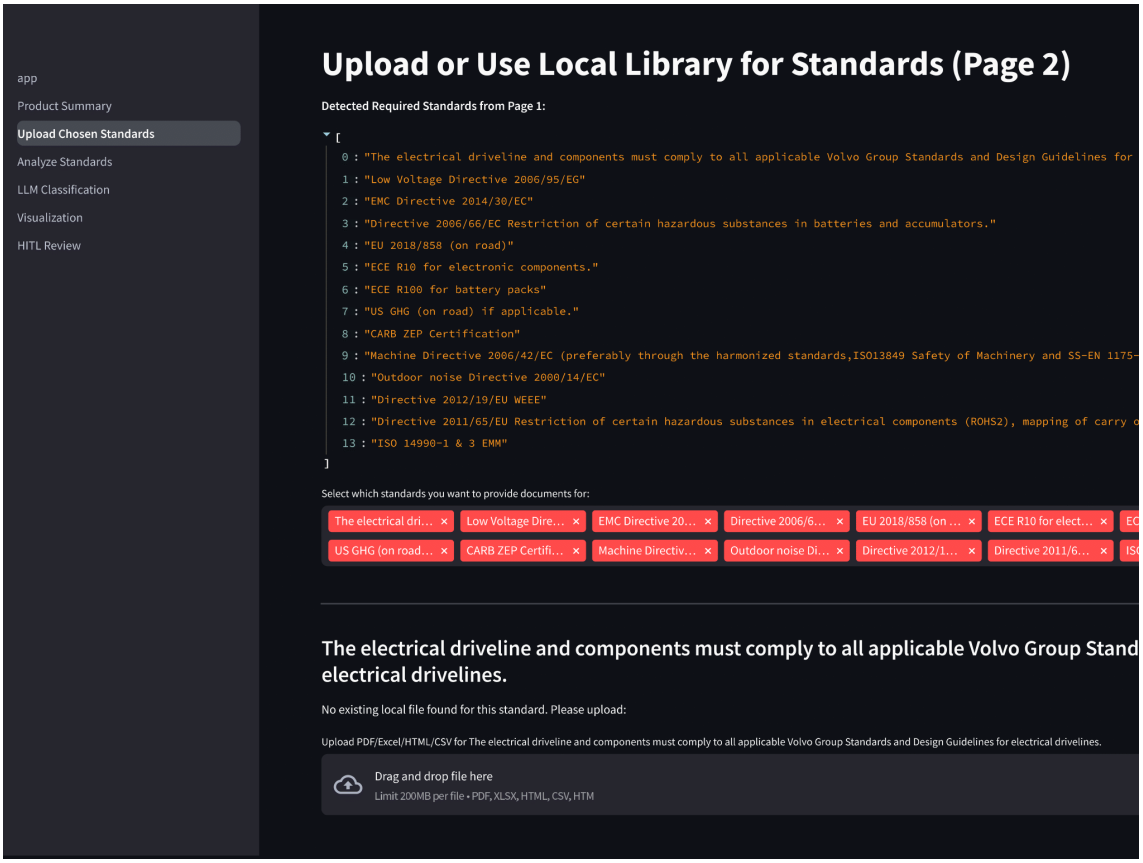

Figure F.5: Page 2 – Upload interface for standards.

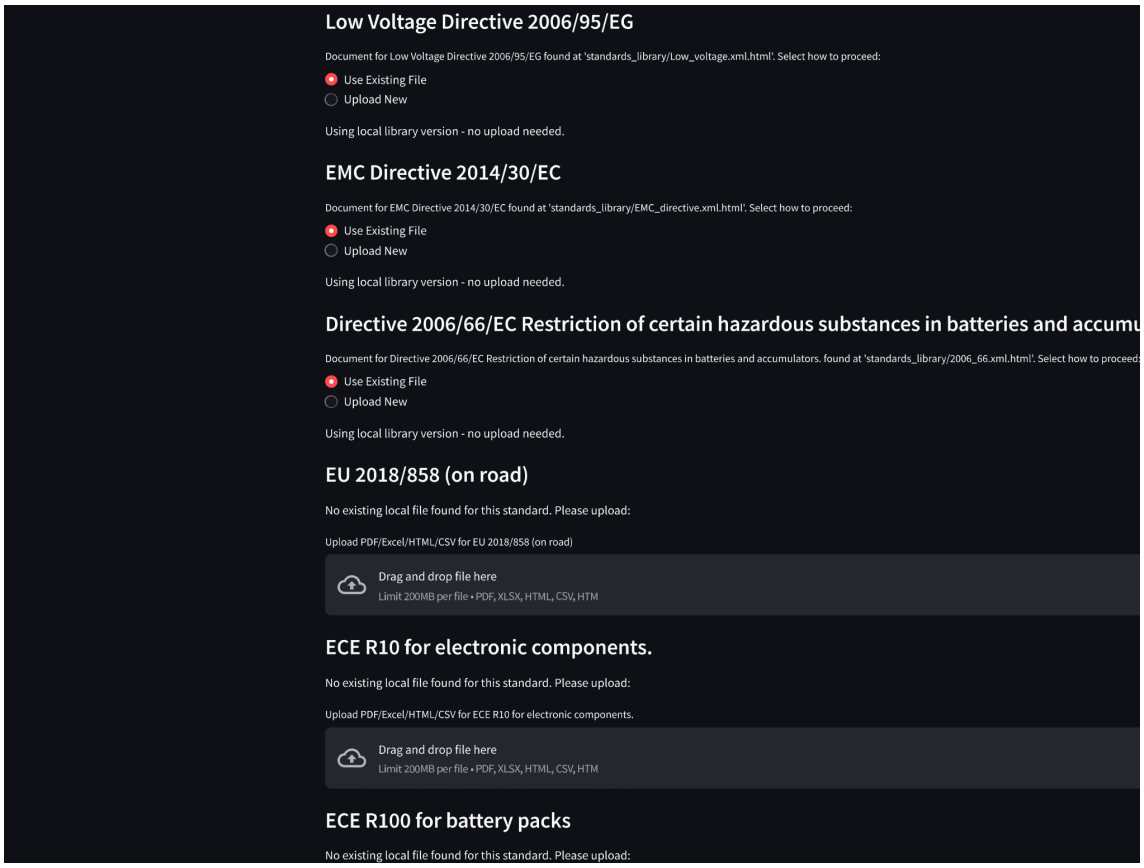

Figure F.6: Page 2 – Standards database mapping interface.

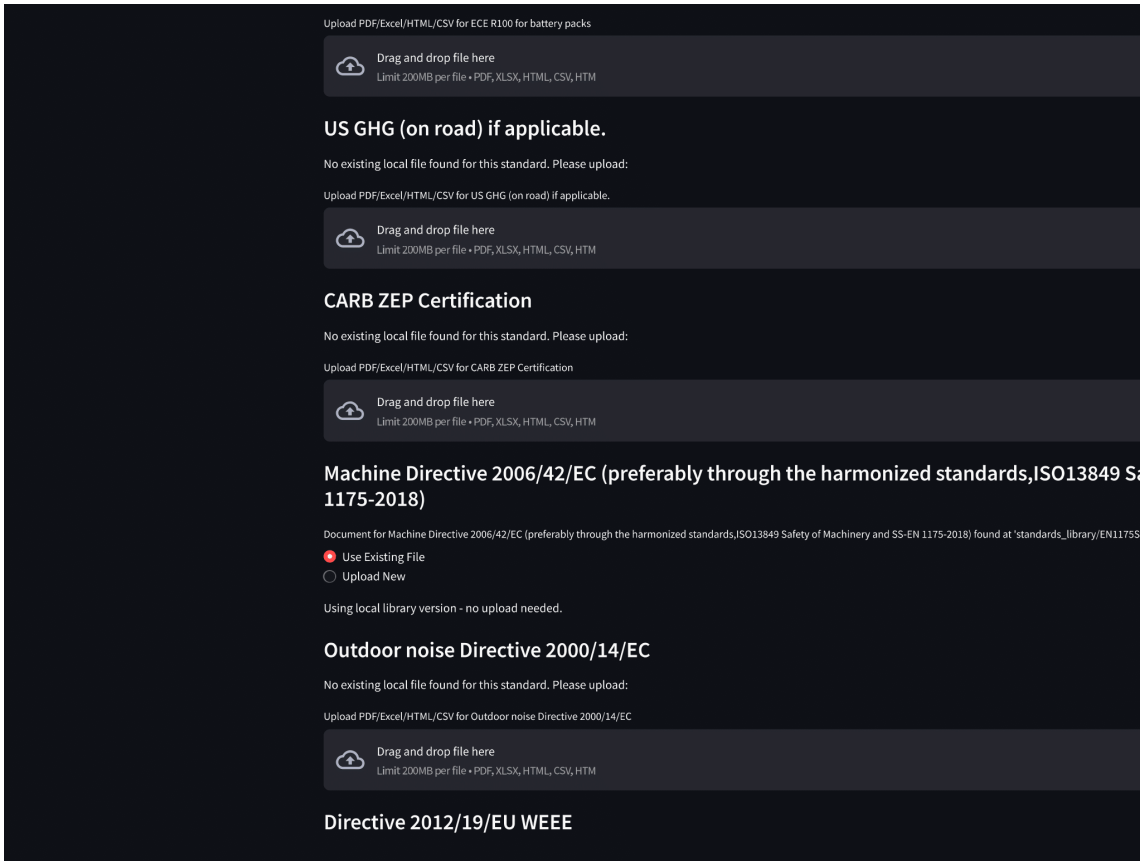

Figure F.7: Page 2 – Standard metadata fields auto-filled.

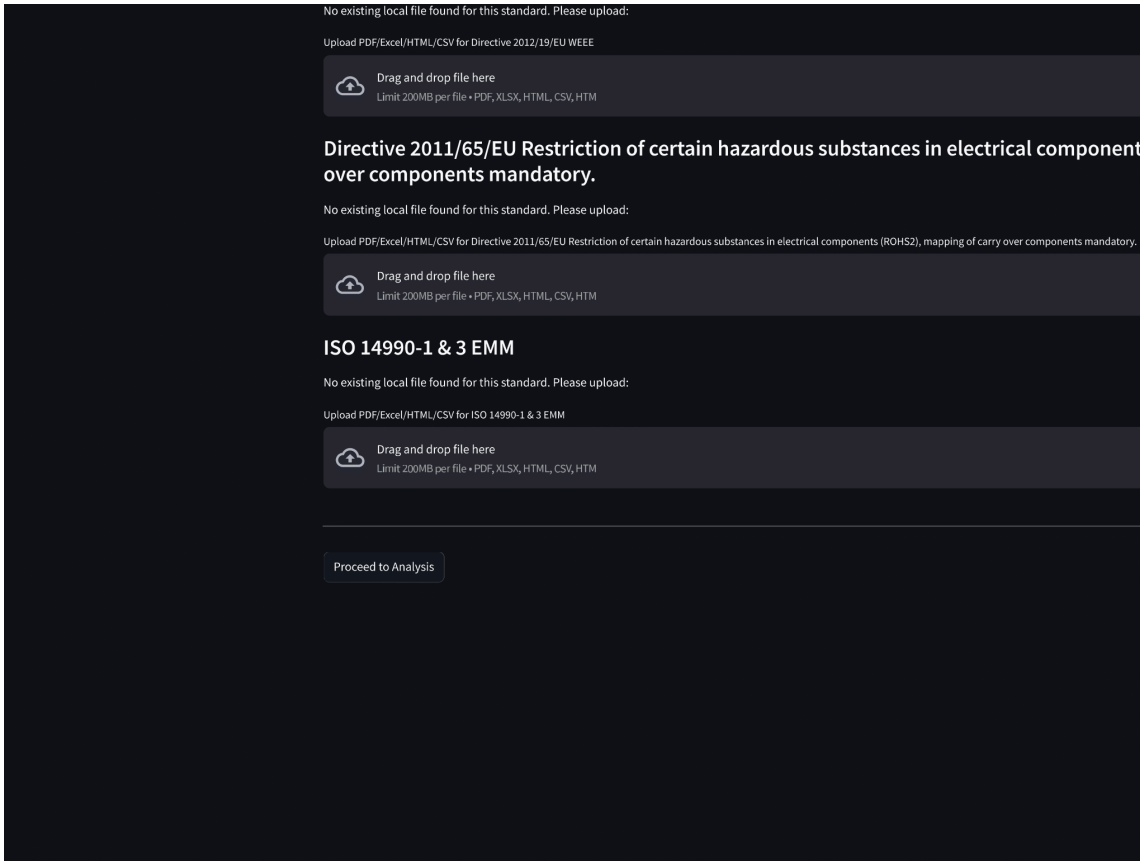

Figure F.8: Page 2 – Proceed to analysis confirmation.

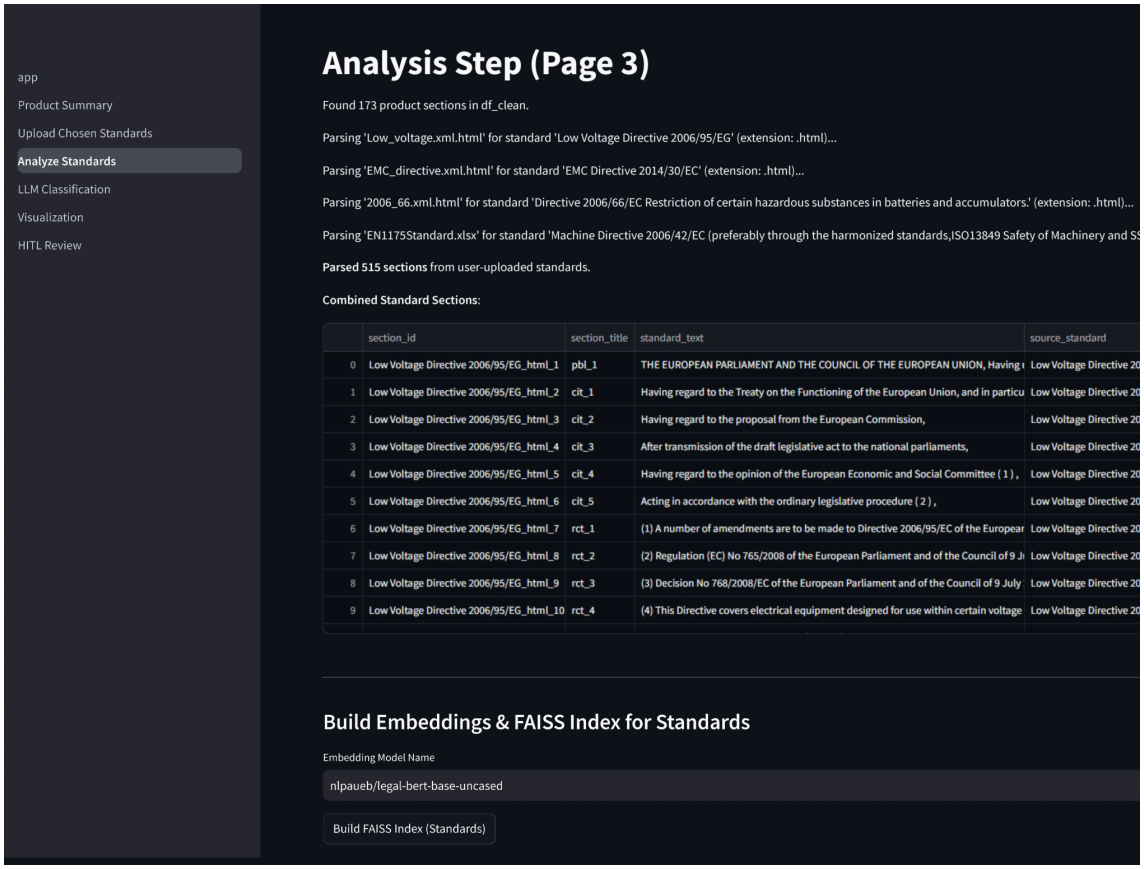

Figure F.9: Page 3 – Embedding configuration and model selection.

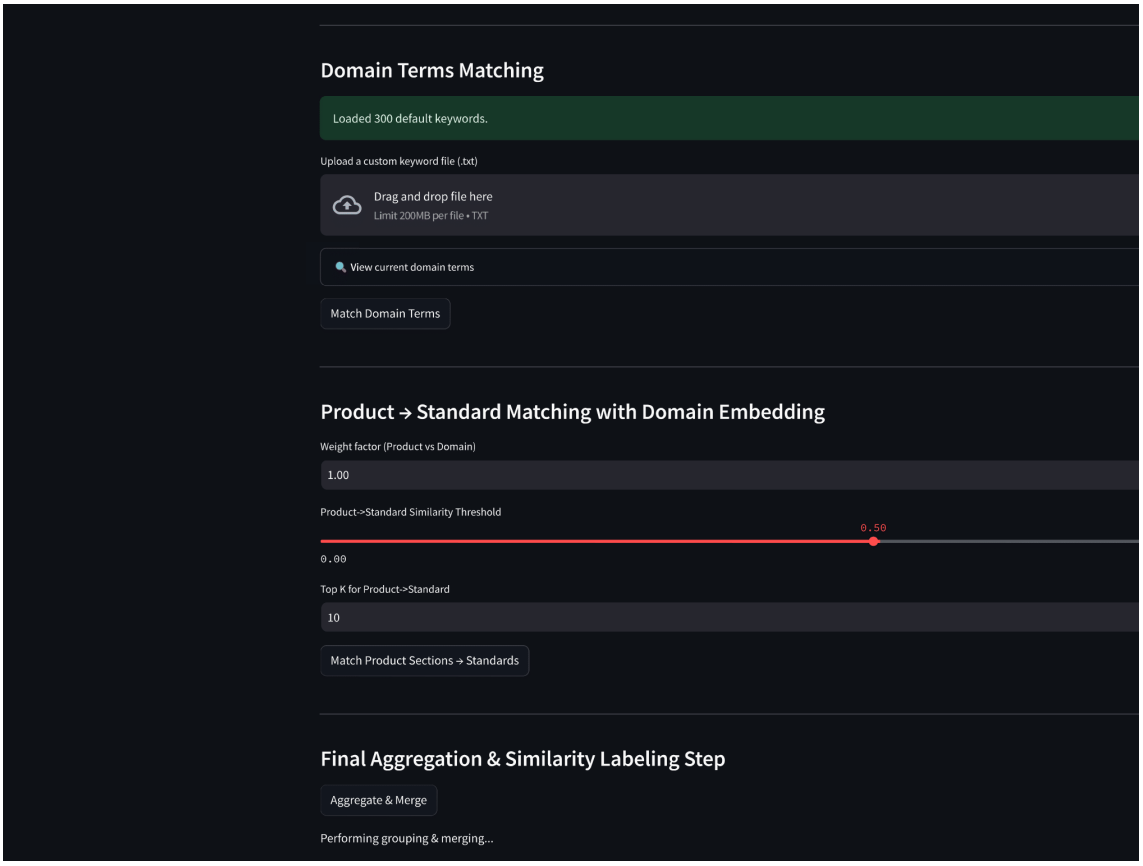

Figure F.10: Page 3 – Domain term matching results.

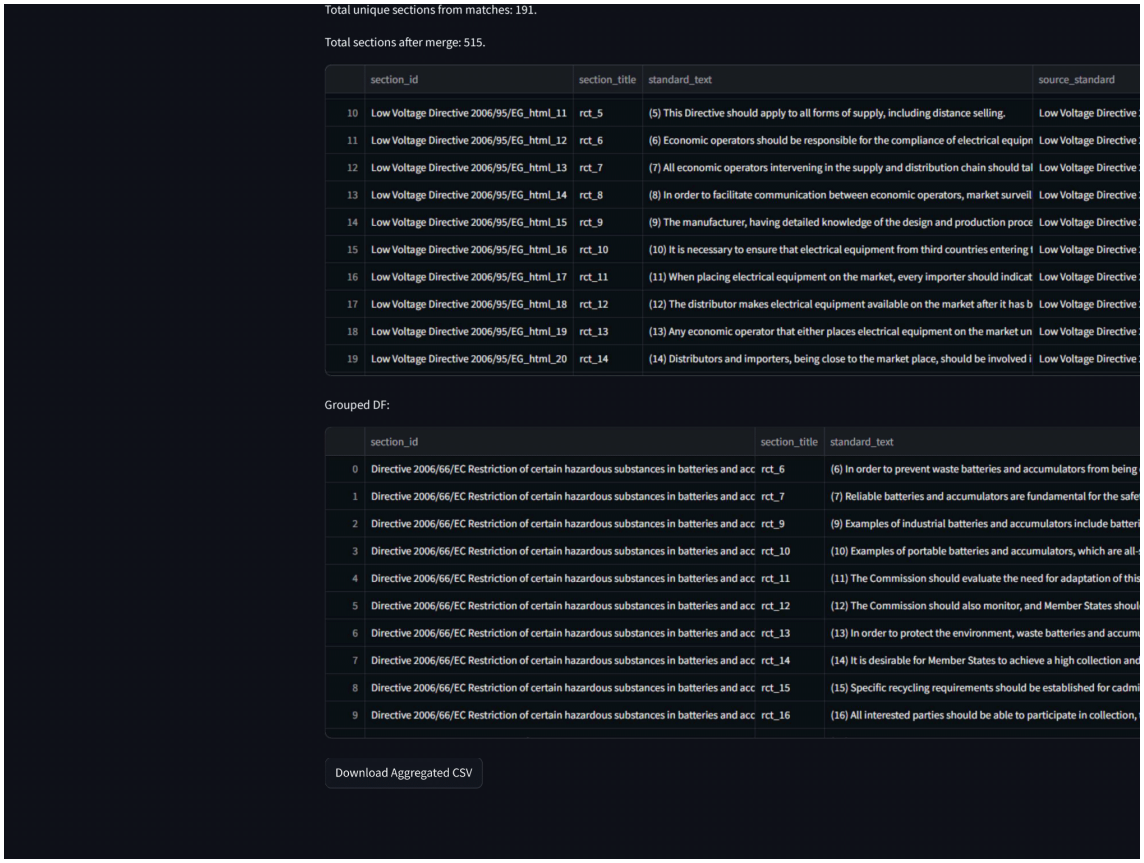

Figure F.11: Page 3 – Matching thresholds and export options.

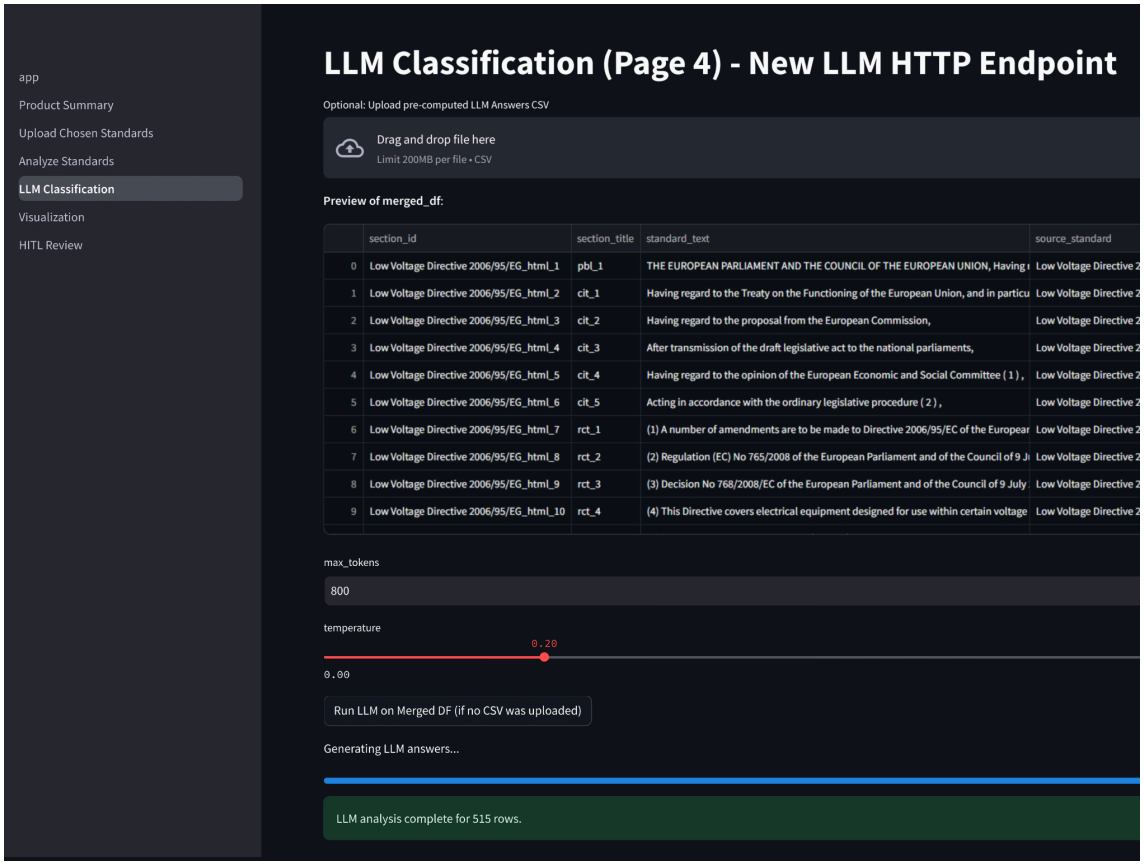

Figure F.12: Page 4 – LLM classification interface with clause preview.

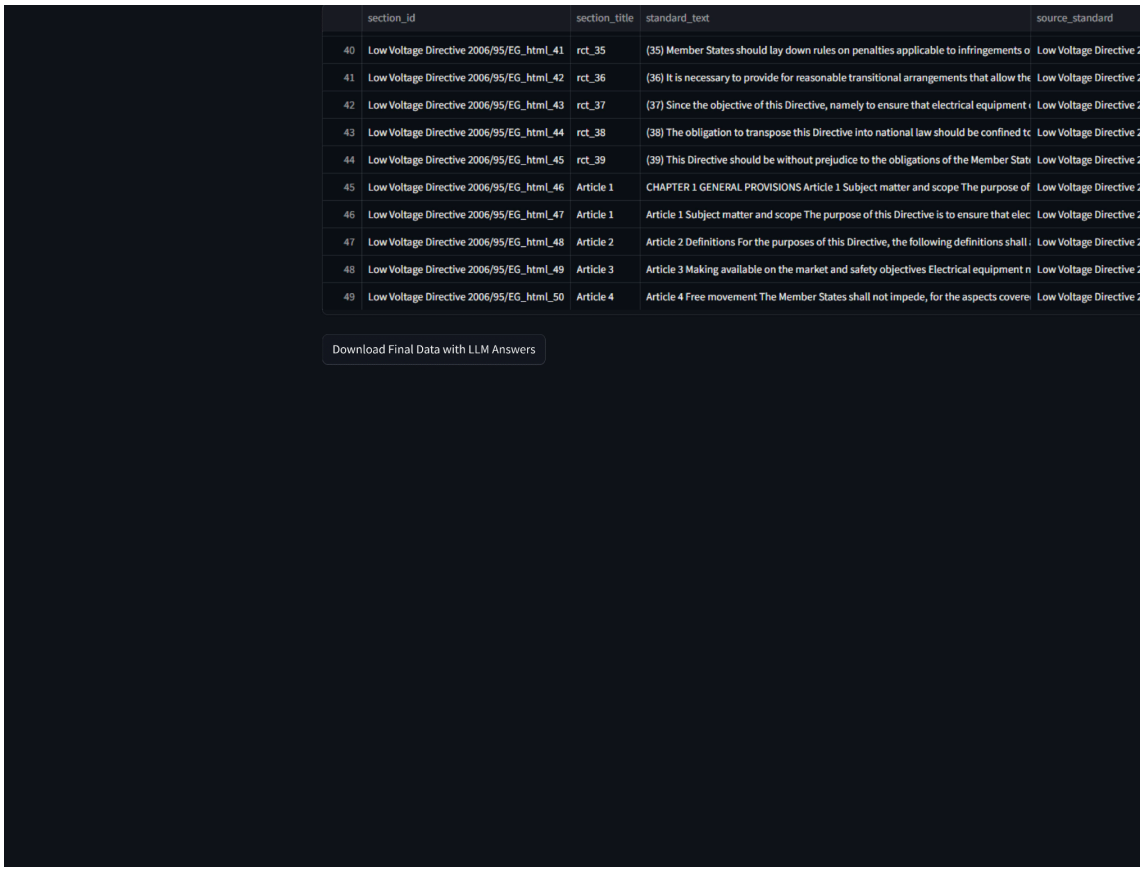

Figure F.13: Page 4 – Justification view for selected clauses.

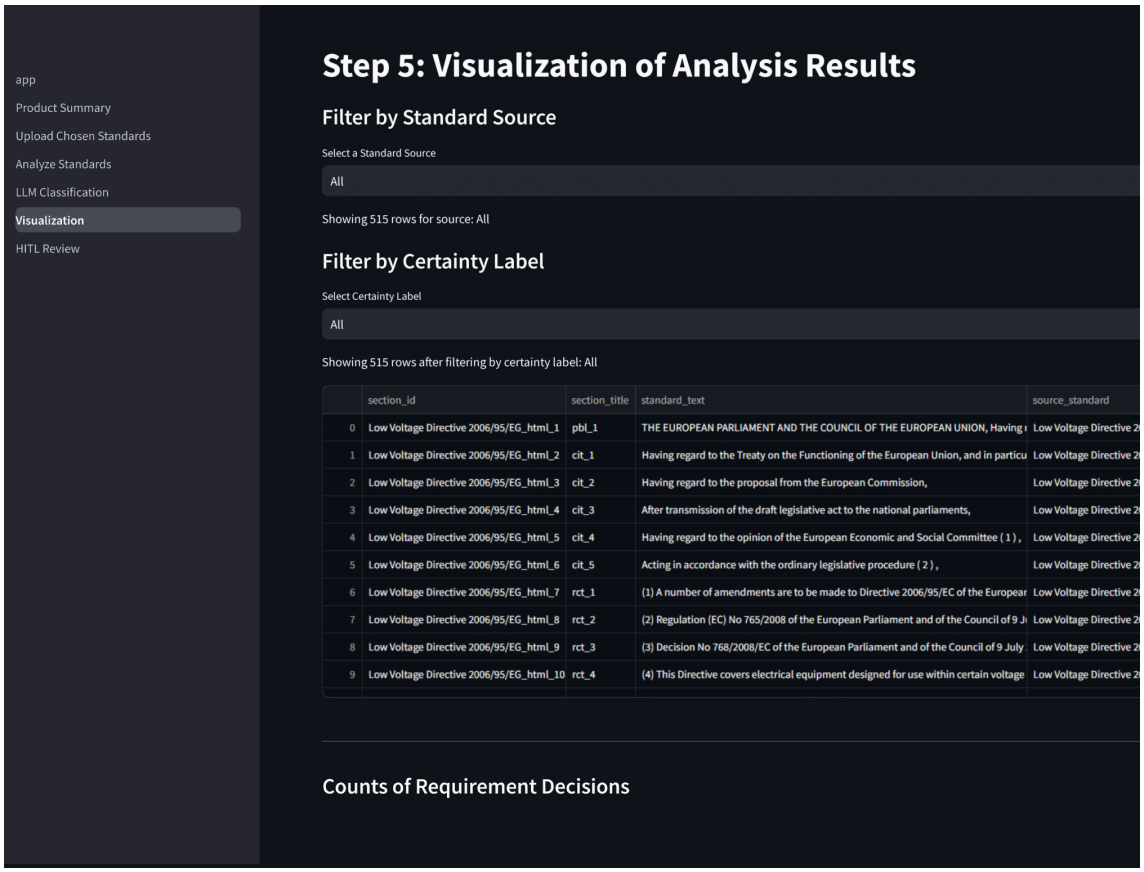

Figure F.14: Page 5 – Visual summary of label distribution.

Figure F.15: Page 5 – Distribution bar chart (expanded view).

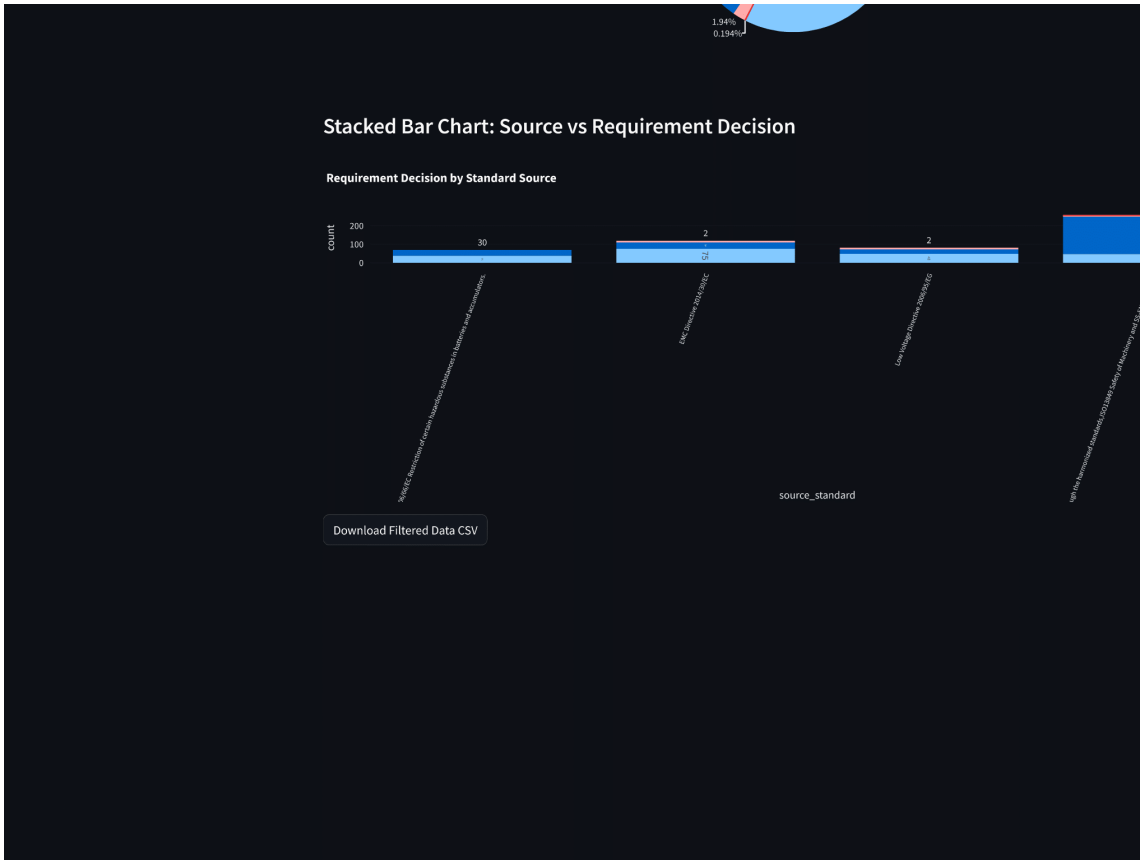

Figure F.16: Page 5 – Clause-level breakdown table.

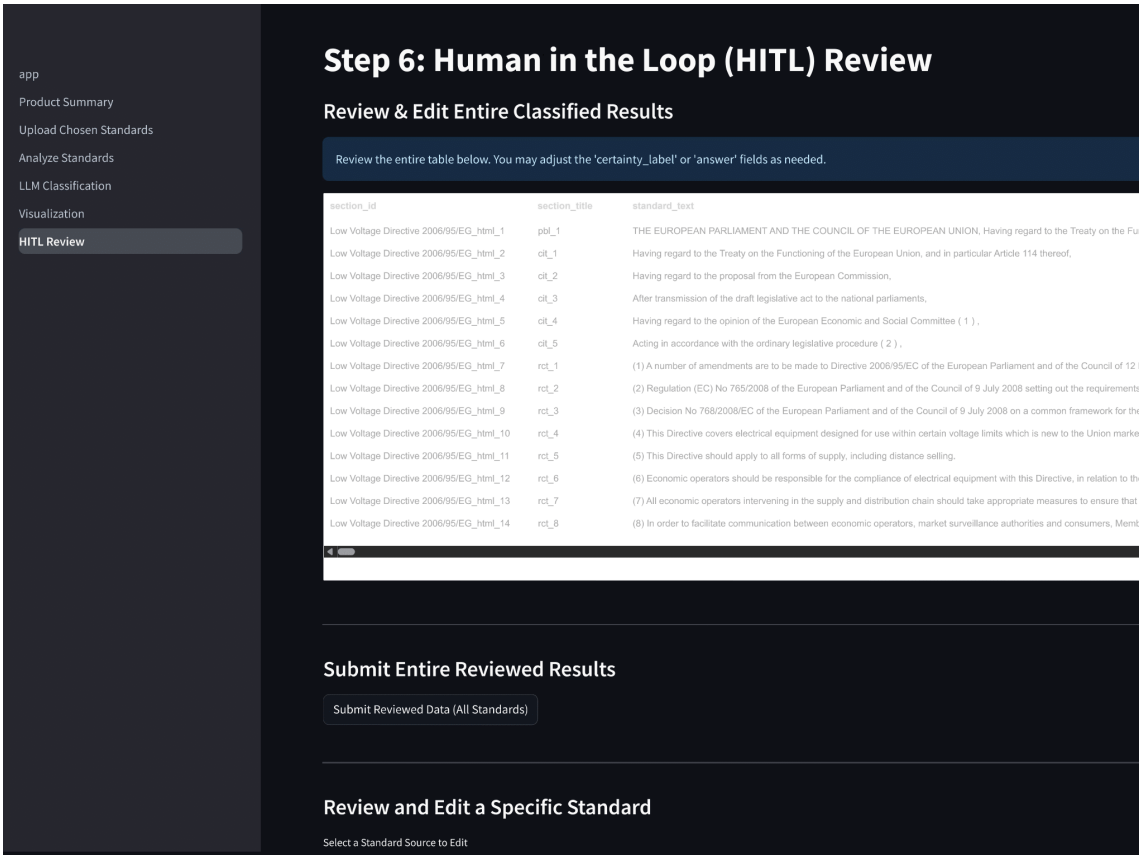

Figure F.17: Page 6 – Human-in-the-loop review interface (HITL).

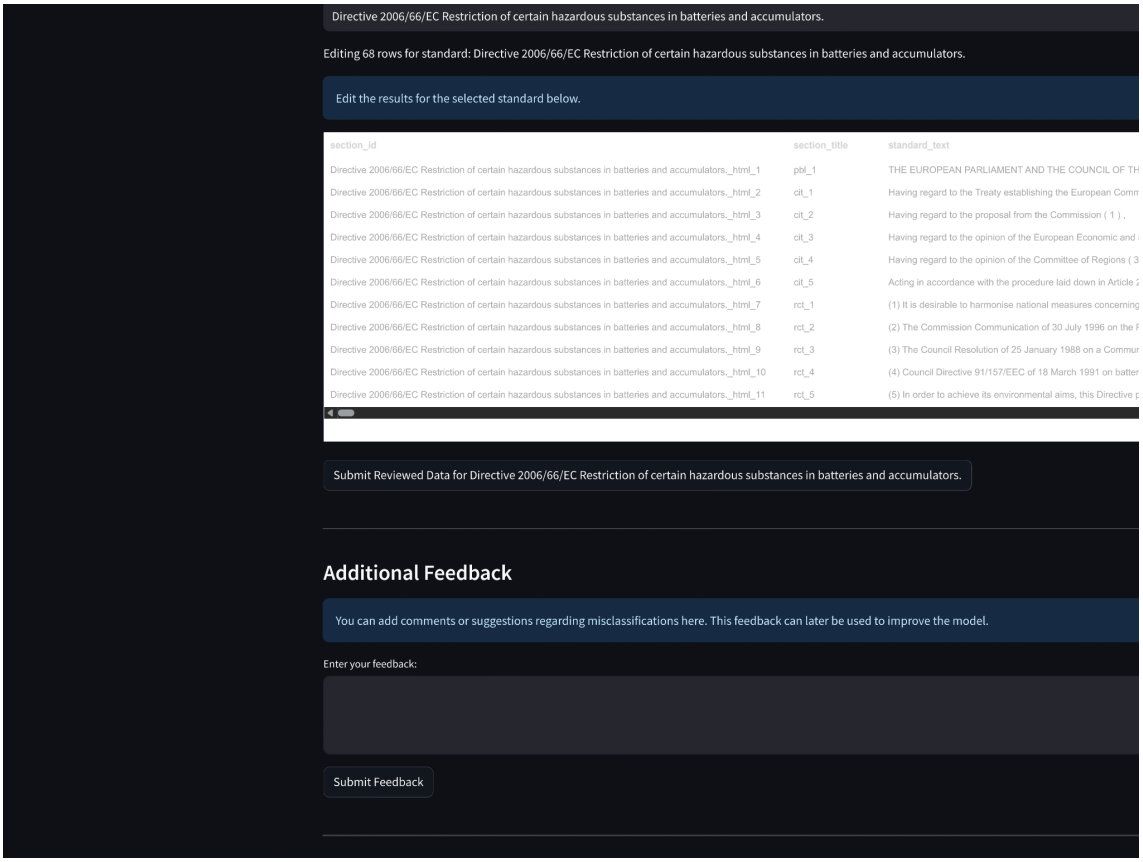

Figure F.18: Page 6 – Manual editing of clause justifications.

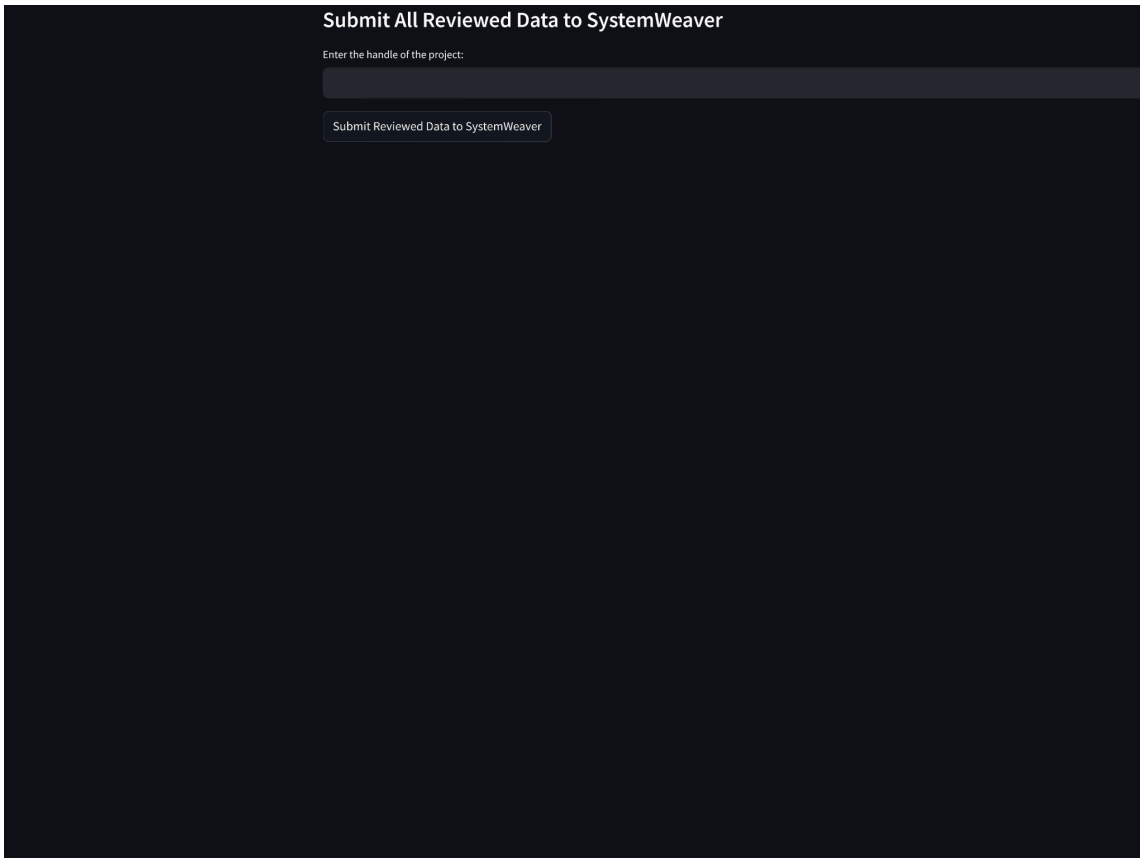

Figure F.19: Page 6 – Export results to SystemWeaver.

G

Interview Protocol

This appendix presents the structured interview guide used in the qualitative evaluation of the prototype. It includes core questions, optional probes, and follow-ups aimed at understanding user experience, perceived accuracy, integration value, and workflow impact.

Interview Questions and Probes

#	**Core Question**	**Optional Probes / Follow-ups**
	Warm-up	
1	When you review a new product for regulatory compliance today, what does a typical session look like?	What information do you consult first? Who else is involved?
2	Before seeing our prototype, how would you rate the pain or time-consumption of that process on a 1–10 scale?	What makes it painful or easy?
	Perception of the Model Output	
3	Looking at the automatically generated clause labels—'Not a Requirement', 'Not Applicable', 'Applicable', how accurate do they feel overall?	Which label appears most reliable? Any obvious misclassifications?
4	For clauses marked 'Applicable', how useful are the extracted obligations and parameter ranges?	Missing details? Too verbose / too brief?
5	Does the certainty colour-coding (Very Certain Relevant, Possibly Relevant, Very Certain Not Relevant, No Match) succeed in directing your attention?	Would you trust low-confidence items? Any alternative signal you'd prefer?

6	Do you think the reasoning process behind the answer is reasonable (e.g., highlighted sentences, certainty scores, parameter extraction)?	Does the explanation feel step-by-step and trustworthy? Why or why not?
Impact on Workflow		
7	If this tool were available tomorrow, would you use it? Where in your workflow would it fit?	Early triage or final checklist? Team or individual use? Time savings estimate?
8	Which manual tasks would disappear or shrink?	Searching for cross-references? Copy-pasting into reports?
Wrap-up		
9	One feature you would add or change to make the tool adoption-worthy—what is it?	Interface change? Explanations? Export format?
10	Would you feel comfortable pressing a single "Generate report" button and not reviewing the results?	Why or why not?

Participant Eligibility Criteria

Table G.2: Minimum participant requirements for interview inclusion.

Criterion	**Minimum Requirement**	**Rationale**
Role Distribution	2 × Compliance/Regulatory Engineers, 2 × System Engineers, 2 × Project or Product Managers	Ensures diverse perspectives across technical, regulatory, and strategic functions.
Project Experience	more than 3 years in discipline *or* atleast 1 full product life-cycle	Participants must have experienced at least one compliance gate from start to finish.
Language Proficiency	Fluent in spoken English	Interviews and prototype interface were conducted in English.

H

Full LLM Justification Example

This appendix contains the complete LLM-generated output for the clause shown in Table 5.5. The response includes structured reasoning and extracted requirements based on both the clause text and matched product content.

Assessment

Requirement Applicable

Reasoning

This section is a requirement because it uses prescriptive language with "shall" statements that impose specific constraints on electrical systems. The section states that "Safety shall not be compromised at any voltage level" and "Electrical systems of trucks powered by lead-acid batteries shall be designed" to operate within specific voltage ranges.

This requirement is applicable to Penta's project for several reasons:

1. The project description explicitly mentions compliance with "SS-EN 1175-2018" which is the standard this clause comes from (EN 1175:2020 is likely an updated version).

2. The project involves electrical drivelines and components, including battery systems, which directly relates to the voltage requirements specified in this clause.

3. The matched engineering keywords include "voltage," "over voltage," "under voltage," and "Power" which align perfectly with this requirement about voltage ranges.

4. Penta's product application data shows they work with "Full electric" and "Electric hybrid" systems, as well as "Energy storage" which would be subject to these voltage requirements.

5. The project description mentions various applications including industrial mobile equipment, which would be covered by this standard

for industrial trucks.

Detailed Requirements

- **Subject**: Electrical systems of trucks powered by batteries
- **Specific Requirements**:
  1. Safety shall not be compromised at any voltage level that can occur.
  2. For lead-acid batteries: Electrical systems shall be designed to operate in the voltage range from 70% up to 120% of the nominal battery voltage.
  3. For other energy source technologies: These voltage limits shall be adapted by the manufacturer.
- **References**: The note mentions that limits set by the manufacturer for other energy sources are outside the scope of this document.