Thesis Dissertation

Automatic Extraction of Privacy Requirements from Natural Language Software Documentation and Alignment with Privacy Legislation

Georgia Georgiou

University of Cyprus

Department of Computer Science

UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

Automatic Extraction of Privacy Requirements from Natural Language Software Documentation and Alignment with Privacy Legislation

Georgia Georgiou

Supervisor
Georgia Kapitsaki

I declare that this dissertation and any accompanying software are my own original work, conducted in full compliance with the University of Cyprus Code of Conduct for Research and with full accountability on my part for the accuracy, integrity, and validity of all content.

May 2026

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Professor Georgia Kapitsaki, for her valuable guidance, continuous support and feedback throughout the development of this research.

Special thanks go to my family and friends for their patience, understanding and constant support during my studies and throughout the preparation of this dissertation. Their emotional support has been my foundation throughout this journey.

Above all, I would like to dedicate this dissertation to my beloved father, whose love and guidance helped shape the person I am today. I hope I have made you proud.

Finally, I would like to acknowledge everyone who contributed directly or indirectly to this work and supported me during the completion of my degree.

Abstract

Software documentation frequently contains requirements related to user privacy and regulatory compliance. However, automatically identifying privacy requirements from natural language remains a difficult task due to ambiguous terminology, implicit privacy meaning, and the overlap between privacy and security concepts.

This thesis presents a hybrid privacy analysis pipeline for the automatic detection of privacy requirements and their alignment with privacy legislation. The proposed system operates in two stages. The first stage identifies privacy-related requirements using transformer-based models combined with rule-based refinement mechanisms. The second stage aligns detected requirements with relevant GDPR articles and CCPA user rights using a multi-label classification approach based on semantic similarity, machine learning, and rule-based techniques.

A web-based application was developed to support interactive privacy analysis and visualization of detected privacy rights. Experimental results demonstrate that the proposed approach can effectively support privacy-aware software engineering and compliance analysis, while also highlighting the challenges introduced by semantic ambiguity and overlapping legal concepts.

Table of Contents

Chapter 1 - Introduction

1.1 General Idea

The rapid growth of information systems and web-based applications has led to the extensive collection and processing of personal data. As a result, privacy protection has become an important concern in modern software engineering. Privacy requirements define the constraints and functionalities that must be incorporated into software systems to ensure the proper handling of personal data and compliance with data protection regulations.

At the same time, privacy regulations such as the General Data Protection Regulation [3] (GDPR) and the California Consumer Privacy Act (CCPA) [5] define various legal obligations and user rights related to personal data processing. Consequently, developers need tools capable of assisting the analysis of privacy requirements and their alignment with legal frameworks.

This thesis focuses on the automated extraction and legal alignment of privacy requirements from natural language software documentation. The proposed approach combines natural language processing techniques, machine learning models, transformer-based architectures and rule-based mechanisms to support the automatic analysis of privacy-related requirements and their association with relevant GDPR articles and CCPA user rights.

In addition, a web-based application was developed to allow users to upload software requirement documents and visualize the generated analysis results interactively.

The overall objective of this work is to provide a framework capable of supporting privacy software engineering and regulatory compliance analysis.

1.2 Motivation

The increasing complexity of modern software systems has significantly expanded the amount of personal data processed by applications and online services. As organizations become more dependent on data-driven functionality, ensuring compliance with privacy regulations has become both a technical and legal challenge.

Although regulations, such as the GDPR and CCPA, provide detailed legal frameworks for data protection, translating these legal obligations into software requirements remains largely a manual process. Developers and analysts must examine requirement specifications, identify privacy statements and determine which legal principles or user rights are relevant to each case. This process is often difficult because privacy concepts are not always explicitly stated within requirement descriptions.

In many cases, requirements contain implicit privacy semantics. Furthermore, privacy terminology frequently overlaps with security-related terminology. These difficulties can lead to incorrect requirement analysis, overlooked privacy obligations and increased compliance risks during software development.

Existing approaches in privacy requirements engineering often focus on isolated tasks, such as privacy requirement detection, requirements extraction, taxonomy construction or legal compliance analysis, without providing an integrated solution that combines automatic privacy identification with regulatory alignment and legal rights mapping [1][2][7][13]. Consequently, there is a growing need for systems capable of automatically analysing software requirements while also associating them with relevant legal concepts and regulatory provisions.

The motivation behind this thesis was the growing complexity of privacy compliance in software systems and the need for automated approaches capable of supporting the analysis of privacy requirements.

1.3 About the Approach

To address these challenges, this thesis proposes a multi-stage automated privacy analysis pipeline for processing natural language software requirements.

The proposed approach operates in multiple stages. Initially, requirements are extracted and preprocessed from textual inputs using natural language processing techniques. The extracted requirements are then analysed by a privacy classification layer responsible for distinguishing privacy-related requirements from non-privacy-related requirements.

Following privacy detection, the identified privacy-related requirements proceed to a second analysis stage where they are semantically aligned with relevant GDPR articles and CCPA user rights. This stage is formulated as a multi-label classification problem, since a single requirement may correspond to multiple legal concepts simultaneously.

The system combines semantic similarity techniques, machine learning models and rule-based heuristics to associate each requirement with relevant GDPR articles and CCPA user rights.

In addition to the analysis pipeline, a dedicated web application was developed to allow users to upload requirement documents and visualize the generated results interactively. The application supports requirement analysis, user validation of predictions, structured presentation of legal concepts and export of results in CSV format. A user validation mechanism was additionally incorporated into the workflow, allowing users to confirm or modify the system predictions before the final stage. This functionality supports greater transparency and provides opportunities for future improvement of the system through feedback collection.

The proposed system aims to facilitate the analysis of privacy requirements and assist developers in aligning software specifications with data protection regulations.

1.4 Structure of the Thesis

The remainder of this thesis is organized as follows.

Chapter 2 presents the background related to privacy requirements engineering and data protection regulations. The chapter discusses the role of privacy in software requirements and provides an overview of the GDPR and the CCPA. It additionally examines how user rights can be associated with software requirements.

Chapter 3 reviews existing research related to automatic privacy requirement detection, natural language processing techniques, large language model approaches and privacy requirements engineering tools. The chapter also discusses the limitations of existing approaches and identifies challenges that motivate the proposed work.

Chapter 4 describes the construction and annotation of the datasets used throughout the thesis. It presents the data collection process, preprocessing and cleaning procedures, annotation methodology, handling of annotation disagreements and the creation of the additional dataset for user rights and article matching.

Chapter 5 presents the proposed experimental methodology and overall system design. The chapter introduces the research pipeline and describes the different approaches explored for privacy classification and legal rights mapping, including classical

machine learning models, transformer-based architectures, prompt-based approaches, and hybrid methodologies.

Chapter 6 presents the experimental evaluation and analysis of results. It discusses the performance of the proposed approaches for both privacy classification and user rights mapping.

Chapter 7 describes the implementation of the developed web-based application and presents its main functionality and workflow. The chapter additionally explains how the proposed pipeline was integrated into the application.

Chapter 8 presents the user evaluation methodology and discusses the results obtained from participant questionnaires and feedback. The chapter additionally analyses user observations regarding the usability and effectiveness of the proposed system.

Finally, Chapter 9 summarizes the conclusions of the thesis, discusses the limitations of the proposed work, and outlines possible directions for future research and system improvements.

Chapter 2 - Background and Legal Framework

2.1 Privacy in Software Requirements

Privacy requirements are a subset of non-functional requirements that define how a system should handle personal data in a secure, lawful and transparent manner. They specify constraints and system behaviours related to the collection, processing, storage and sharing of personal information, ensuring that users' privacy is preserved throughout the system lifecycle [1].

In software engineering practice, requirements are typically expressed in natural language, either as formal specifications or as user stories. For example, statements such as "The system shall allow users to delete their personal data" or "As a user, I want to control how my data is used" reflect privacy-related concerns. However, requirement specifications written in natural language may contain ambiguities and inconsistencies, which make the identification of non-functional requirements such as privacy particularly challenging [2].

Privacy requirements may be either explicitly stated or implicitly embedded within system specifications. Explicit requirements directly refer to privacy-related concepts. In contrast, implicit privacy requirements do not explicitly mention privacy but can be inferred from the context of the requirement, often requiring semantic analysis to be identified [2].

The identification of privacy requirements, whether explicit or implicit, is essential for ensuring compliance with data protection regulations. However, automatically detecting such requirements remains a complex task, particularly when dealing with large volumes of unstructured textual data [2].

In the context of this thesis, the distinction between explicit and implicit privacy requirements is particularly important, as the proposed system aims to identify both types using a combination of machine learning and rule-based techniques.

2.2 GDPR Overview

The General Data Protection Regulation (GDPR) is a comprehensive data protection framework that governs the processing of personal data within the European Union. It

establishes a set of principles and rights aimed at safeguarding individuals' privacy while ensuring transparency, accountability and lawful data processing practices [3].

Instead of providing an exhaustive analysis of all GDPR provisions, this work focuses on the concepts that are most relevant to software requirements engineering. Core principles such as data minimization, purpose limitation, transparency, as well as integrity and confidentiality play a crucial role in shaping how systems should manage personal data. These principles influence how systems should manage personal data and provide a basis for defining privacy-related system requirements [1].

In addition to its core principles, the GDPR defines a set of fundamental user rights that influence the handling and protection of personal data within modern systems and organisations. These rights include obligations related to transparency, data access, correction, deletion, portability and control over personal data processing activities. Table 1 presents some of the main GDPR rights and articles [3].

GDPR Article	***User Right***	***Short Description***
Article 13–14	Right to Information	Users must be informed about data collection and processing
Article 15	Right of Access	Users can access their personal data
Article 16	Right to Rectification	Users can correct inaccurate personal data
Article 17	Right to Erasure	Users can request deletion of personal data
Article 18	Right to Restrict Processing	Users can limit data processing activities
Article 20	Right to Data Portability	Users can receive and transfer their data
Article 21	Right to Object	Users can object to certain processing activities
Article 22	Rights related to Automated Decision-Making	Protection against solely automated decisions

Table 1: Overview of Main GDPR User Rights

From a software engineering perspective, these rights can be operationalized into concrete system functionalities. For example, the right to erasure implies the need for mechanisms that support secure and complete data deletion, while the right of access requires systems to provide user interfaces or APIs through which individuals can retrieve their personal data. Consequently, GDPR provisions can be interpreted as a

source of high-level requirements that must be translated into implementable system features [3][4].

While GDPR defines several principles for lawful data processing, this thesis focuses primarily on user rights, as they provide a direct and operational link between legal requirements and system functionality. Unlike general principles, user rights can be explicitly translated into actionable system behaviours, such as allowing users to access, modify or delete their personal data. For this reason, user rights constitute the primary target of the analysis performed in this work.

2.3 CCPA Overview

The California Consumer Privacy Act (CCPA) is a data privacy law that grants consumers in California specific rights over their personal information. While sharing similar objectives with the GDPR, the CCPA places greater emphasis on consumer control, particularly in relation to data collection practices and the sharing or sale of personal information [5].

The CCPA defines a set of fundamental consumer rights that influence the handling and protection of personal information by organizations. These rights include transparency regarding the collection and use of personal information, the ability to request access to or deletion of personal data, the right to opt out of the sale or sharing of personal information and protection against discrimination for exercising privacy rights. Table 2 presents some of the main CCPA rights and provisions [5].

CCPA Right	***Short Description***
Right to Know	Consumers can know what personal information is collected and how it is used
Right to Delete	Consumers can request deletion of personal information
Right to Access	Consumers can access collected personal information
Right to Opt-Out	Consumers can opt out of the sale or sharing of personal information
Right to Non-Discrimination	Consumers cannot be discriminated against for exercising privacy rights
Right to Limit Use of Sensitive Personal Information	Consumers can restrict the use of sensitive personal data

Table 2: Overview of Main CCPA Rights

These rights translate into both functional and non-functional requirements within software systems. For instance, systems must support transparency by providing clear information about data usage, enable user-driven actions such as deletion requests and incorporate mechanisms that allow users to manage their data-sharing preferences. The provisions of the CCPA therefore influence system design and can be used to derive privacy-related requirements [1].

Furthermore, the rights defined in the CCPA are particularly relevant for software systems, as they can be translated into functional requirements. In this work, these rights are treated as operational categories that can be detected from natural language requirements and used to assess compliance.

2.4 Mapping User Rights to Software Requirements

A central challenge in privacy-aware software engineering lies in relating user rights defined in legal frameworks to concrete system specifications. Regulatory frameworks, such as the GDPR and the CCPA, define user rights in an abstract and legal-oriented manner, whereas software systems require precise and well-defined requirements.

In this work, user rights are interpreted as entities that can be operationalized into software requirements. More specifically, each legal right can be mapped to one or more system functionalities. For instance, the right of access can be translated into mechanisms that allow users to view or retrieve their personal data, the right to deletion requires functionality for securely removing stored information and the right to opt out necessitates controls for managing data-sharing preferences. Through this perspective, legal concepts are transformed into implementable features within software systems.

Establishing such a mapping enables privacy requirements to be represented as identifiable categories within textual data. This representation is particularly important for enabling automated analysis, as it allows machine learning and natural language processing techniques to detect and classify privacy-related requirements [2].

Furthermore, this mapping constitutes a fundamental component of the proposed system. It forms the basis of the second stage of the pipeline, in which requirements identified as privacy-related are semantically aligned with specific legal concepts or user rights. By linking textual requirements to regulatory provisions, the system

facilitates a deeper understanding of how software specifications relate to legal obligations.

In the context of this thesis, user rights are treated not only as legal concepts but also as classification targets within the proposed system. Each requirement identified as privacy-related is mapped to one or more user rights, effectively transforming legal obligations into machine-detectable categories. This perspective enables the application of machine learning and NLP techniques to bridge the gap between unstructured software requirements and structured legal knowledge.

Ultimately, the ability to automatically connect software requirements to data protection regulations can support developers in designing privacy-compliant systems and assist organizations in evaluating and auditing their software artifacts. In this way, the proposed approach contributes to bridging the gap between legal frameworks and practical software engineering processes.

Chapter 3 - Related Work

3.1 Automatic Detection of Privacy Requirements

Recent research has increasingly focused on the automatic detection of privacy requirements from textual software artifacts, with Natural Language Processing (NLP) and machine learning techniques widely adopted to identify privacy-related content in natural language requirements.

Among these approaches, a representative method is proposed by Casillo et al. [2], who investigate the automatic detection of privacy requirements from user stories using NLP and deep learning techniques. User stories, which are widely used in Agile development, typically follow a structured format (e.g., "As a [role], I want to [feature], so that [reason]"), yet they often contain implicit or ambiguous references to privacy aspects, which complicates the automatic identification of privacy-related requirements.

To address this problem, the authors propose a method that combines linguistic feature extraction with deep learning models. Specifically, NLP techniques are used to capture both syntactic and semantic characteristics of the text, including tokenization, part-of-speech tagging, dependency parsing and named entity recognition. These features are complemented by lexicon-based information derived from privacy dictionaries, which capture domain-specific privacy terminology.

The extracted features are then used to train convolutional neural networks (CNNs) for binary classification, distinguishing between user stories that contain privacy-related content and those that do not. In addition, the authors introduce a transfer learning strategy, where a pre-trained neural network trained on privacy disclosure detection tasks is adapted to the user story domain. This approach allows the model to leverage knowledge from larger datasets and improve performance in scenarios with limited labelled data.

The empirical evaluation, conducted on a dataset of 1,680 user stories, demonstrates that deep learning models outperform traditional machine learning approaches such as Support Vector Machines and Logistic Regression. Furthermore, the use of transfer learning leads to an improvement of approximately 10% in both accuracy and F1-score,

highlighting the effectiveness of transfer learning for privacy-related text classification [2].

While the approach demonstrates strong performance, several limitations still persist. Privacy requirements are often expressed implicitly and may depend heavily on context, which can make their automatic detection difficult even when advanced NLP and deep learning techniques are used. In addition, the approach primarily focuses on identifying whether a requirement is privacy-related, rather than mapping detected requirements to specific legal obligations or user rights defined in regulatory frameworks such as the GDPR or CCPA. Consequently, although such methods can support the identification of privacy-related content, additional techniques are necessary to associate requirements with relevant legal obligations and privacy regulations.

This line of work demonstrates the feasibility of automating privacy requirement detection using NLP and machine learning techniques. At the same time, it highlights the need for more comprehensive approaches that combine requirement detection with higher-level semantic reasoning and legal alignment.

3.2 NLP and LLM-based Approaches

Recent advances in NLP and LLMs have significantly expanded the possibilities for automatically extracting and generating privacy requirements from software-related textual artifacts. A representative example is the work of Baldwin et al., who propose an LLM-based framework for extracting privacy behaviours from software documentation and generating privacy requirements in the form of privacy user stories [6].

Their framework uses prompt engineering techniques, including chain-of-thought (CoT) prompting and in-context learning (ICL), to guide LLMs in identifying privacy-related elements such as actions, data types and processing purposes. These extracted behaviours are organized according to an extended privacy taxonomy and are then used to generate structured “privacy stories”, following a simplified template that describes how an application processes personal data [6].

The authors evaluate several LLMs, including GPT-4o, Llama 3 and Qwen models. Their results show that LLMs can achieve strong performance in privacy behaviour

extraction and privacy story generation, with the best-performing models reaching F1 scores above 0.8 in some evaluation settings [6]. The study also shows that supervised fine-tuning can improve the performance of smaller models, although additional preference-based tuning does not always lead to further improvement and may cause overfitting when the dataset is small [6].

Although the proposed framework demonstrates strong potential, several challenges remain. The authors report that LLMs may produce inconsistent outputs, including hallucinated labels or labels that do not match the predefined taxonomy. In addition, the approach focuses mainly on extracting privacy behaviours and generating privacy stories, rather than explicitly mapping these requirements to legal frameworks such as GDPR or CCPA [6].

This work demonstrates the potential of LLM-based approaches for automating privacy requirements engineering from diverse software artifacts. However, the limitations related to output consistency, predefined taxonomies and lack of explicit legal alignment show that further integration with compliance-oriented methods is still needed [6].

3.3 Automated Privacy Requirement Generation Systems

In addition to approaches focused on detection and extraction, recent research has explored the development of automated systems for generating privacy requirements from software artifacts. These systems aim to support developers by transforming informal requirements into structured privacy-aware specifications.

A representative example is the PrivacyStory tool, which provides an end-to-end pipeline for extracting and generating privacy requirements from user stories [7]. The tool is designed to support agile development environments, where user stories are widely used as a primary form of requirements specification.

PrivacyStory operates through a multi-stage pipeline. Initially, user stories are analysed to ensure syntactic quality, followed by a classification step to identify whether they contain privacy-related content. Subsequently, the system applies Named Entity Recognition (NER) techniques to detect key privacy-related entities, including personal data, data subjects and processing activities. These elements are then used to automatically generate Data Flow Diagrams (DFDs), which model the relationships

between data subjects, personal data, and processing activities. Based on the extracted information, the system generates privacy requirements using predefined patterns derived from established privacy frameworks such as LINDDUN and ProPAN [7].

While PrivacyStory demonstrates the feasibility of automating privacy requirements generation, several limitations remain. The approach relies heavily on predefined templates and structured patterns, which may potentially limit flexibility when dealing with diverse or complex requirements. In addition, the system requires human validation to ensure the correctness and completeness of the generated outputs [7]. Furthermore, the generated requirements are not explicitly linked to legal frameworks such as GDPR or CCPA, limiting their direct applicability for compliance purposes.

Automated systems such as PrivacyStory highlight the potential of pipeline-based approaches for privacy requirements engineering, while also revealing the need for more advanced solutions that integrate automation with semantic understanding and legal alignment.

3.4 Extraction from Legal and Regulatory Texts

Another important research direction focuses on the extraction of privacy requirements directly from legal and regulatory documents, aiming to bridge the gap between high-level legal texts and implementable software requirements.

A foundational contribution in this area is provided by Breaux et al., who explored methods for extracting rights and obligations from legal texts such as HIPAA and refining them into software requirements. Their work demonstrated that regulatory provisions can be systematically analysed and aligned with structured software requirements [8].

Building on this line of research, Breaux and Gordon proposed a legal requirements specification language (LRSL) to codify policy and legal requirements using semi-formal specifications. Their approach also supports the traceability of regulatory requirements across multiple jurisdictions, enabling a more structured analysis of regulatory constraints. [9]

Several works have also explored the automation of requirement extraction using tool-supported and NLP-based techniques. For example, Zeni et al. proposed GaiusT, a tool that applies semantic annotation techniques to semi-automatically extract rights and

obligations from legal documents. The approach can process hierarchical structures and cross-references within legal texts and supports multilingual analysis. [10]

Similarly, Sleimi et al. proposed an automated framework for extracting semantic legal metadata from legal texts using NLP and machine learning techniques. Their approach identifies semantic metadata types including actors, obligations, conditions, exceptions and violations, in order to support the systematic analysis of legal requirements. The framework combines NLP-based parsing techniques with machine learning methods to transform unstructured legal text into structured semantic representations that facilitate legal requirements analysis [11].

More recent approaches have focused on regulatory compliance analysis. Torre et al. developed a conceptual model for characterizing the content of privacy policies according to GDPR provisions and proposed an AI-assisted approach for checking the completeness of privacy policies. Their method identifies metadata types within legal statements and models' dependencies between them in order to support automated completeness and compliance checking [12].

A closely related contribution is provided by Sangaroonsilp et al., who developed a comprehensive taxonomy of privacy requirements derived from GDPR, ISO/IEC 29100, Thailand PDPA and the APEC privacy framework. The taxonomy consists of 71 privacy requirements organized into seven categories, including user participation, notice and security. The authors further validated this taxonomy by mining issue reports from large open-source projects, demonstrating its applicability in real-world software development contexts. Importantly, this work establishes a structured relationship between legal provisions and software-level privacy requirements, supporting compliance-oriented requirements engineering [13].

Despite these contributions, existing approaches still present several limitations. Many methods primarily focus on analysing legal texts and extracting structured information, while offering limited integration with practical software engineering workflows. In addition, several approaches are tailored to specific regulations, domains or application contexts, which may limit their generalizability and broader applicability in real-world software development environments.

3.5 Limitations of Existing Approaches

Despite the significant progress achieved in privacy requirements engineering, the approaches reviewed in Sections 3.1–3.4 still exhibit several important limitations.

NLP and machine learning methods have demonstrated strong performance in detecting privacy-related requirements from textual artifacts. However, as shown by Casillo et al., the automatic identification of privacy disclosures remains difficult due to the wide variety of possible terms and the contextual nature of privacy-related information in user stories [2]. Furthermore, these methods mainly focus on identifying whether a requirement is privacy-related or not, without directly associating the requirement with specific legal concepts.

Recent advances in LLMs have further improved the ability to extract and generate privacy-related information from unstructured software artifacts. Baldwin et al. demonstrated that LLM-based techniques can effectively identify privacy behaviours and generate structured privacy stories [6]. However, the study also highlighted important limitations, including hallucinated labels, inconsistent adherence to predefined taxonomies and difficulties in maintaining structured outputs [6].

Pipeline-oriented systems, such as PrivacyStory, demonstrate the feasibility of automating privacy requirement generation through integrated workflows involving classification, entity extraction and template-based requirement generation [7]. However, these systems rely heavily on predefined patterns and still require human validation to ensure the correctness and completeness of the generated outputs.

Similarly, research focused on extracting information from legal and regulatory texts, including taxonomy-based and semantic analysis approaches, has introduced valuable techniques for identifying rights, obligations, constraints and legal metadata structures [8][11]. However, these approaches are primarily centred on legal document analysis and remain only loosely connected to practical software engineering workflows. Spósito et al. further highlight the need to bridge the gap between theoretical privacy engineering knowledge and real-world implementation, emphasizing the importance of more effective and adaptable privacy-preserving frameworks [1]. Together, these observations suggest that existing approaches still provide limited integration between legal privacy analysis and practical software requirements engineering processes.

Existing research tends to address individual aspects of privacy requirements engineering in isolation, such as privacy detection, information extraction, legal analysis or requirement generation. To the best of our knowledge, no existing approach explicitly combines automated privacy requirement classification with direct mapping to GDPR and CCPA user rights within a unified analysis pipeline.

This limitation is particularly significant in software development, where developers need not only to identify whether a requirement is privacy-related, but also to understand which regulatory implications are associated with that requirement. The absence of integrated end-to-end solutions therefore reduces the practical applicability of many existing approaches in real-world privacy engineering and regulatory compliance workflows.

These limitations highlight the need for approaches that combine privacy requirement detection with legal analysis, allowing developers to better understand the regulatory implications of software requirements.

Chapter 4 - Dataset Construction and Annotation

4.1 Overview of Dataset Construction

The construction of the dataset constitutes a critical component of this thesis, as the performance and reliability of the proposed models depend directly on the quality, diversity and representativeness of the training data. Given the limited availability of publicly accessible datasets specifically tailored to privacy requirements extraction, a hybrid dataset construction approach was adopted. This approach combines multiple data sources to create a comprehensive corpus suitable for training and evaluating machine learning models.

More specifically, the dataset was constructed by combining two primary sources: a publicly available dataset of user stories originally provided by Dalpiaz [14], which has also been used in prior work on privacy requirement detection [2], and a manually constructed dataset inspired by recent research on privacy story generation and privacy behaviour extraction [6].This strategy allows the dataset to capture both general software requirements expressed in natural language and privacy-related requirements, thereby improving the ability of the models to generalize across different contexts.

4.2 Data Sources and Selection Criteria

An extensive search was conducted across academic literature and public repositories, including GitHub, Zenodo and Kaggle, to identify datasets related to privacy requirements. However, the findings revealed that datasets specifically targeting privacy requirements extraction remain extremely limited. Most publicly available datasets focus either on general non-functional requirements or privacy policies, rather than software requirement specifications. As a result, recent research in the field frequently relies on manually annotated or manually constructed datasets for privacy requirements analysis [6].

The primary dataset used in this thesis was derived from the user stories dataset introduced by Dalpiaz [14], which provides a large collection of software requirements expressed in structured natural language. This dataset was selected because user stories represent a widely adopted method for documenting requirements in modern software

development, making them highly relevant for the task of automatic requirement analysis.

In addition to this dataset, a second privacy-oriented dataset presented by Baldwin et al. [6] was incorporated. This dataset consists of manually annotated software documents and privacy stories, providing examples of privacy-related concepts and privacy behaviours relevant to privacy requirements analysis.

The combination of these two datasets enables the creation of a hybrid corpus containing both general software requirements and privacy-related requirements. This is particularly important because privacy requirements are often embedded within broader requirement descriptions and may not always be explicitly stated.

4.3 Composition of the Main Corpus

After integration, the main corpus used for privacy classification consisted of 2,614 requirement statements in total. Of these, 1,680 originated from the User Stories dataset [14], while the remaining 934 were obtained from the second source [6]. The merged dataset was then used as the basis for binary annotation, with each requirement labelled as either privacy-related or non-privacy-related.

During dataset annotation, 843 requirements were manually labelled as privacy-related, while 1,771 were labelled as non-privacy. This distribution indicates a moderate class imbalance, reflecting real-world scenarios where privacy-related requirements are less frequent and often embedded within broader functional specifications. Despite this imbalance, the dataset provides sufficient coverage of both classes to support reliable model training and evaluation.

The objective of this corpus was to support the first stage of the proposed pipeline, which focuses on the automatic identification of privacy requirements from software requirements. Therefore, the dataset includes requirements written in different styles and containing varying levels of privacy-related information, allowing the classification models to learn from both explicit and implicit privacy requirements.

4.4 Preprocessing and Cleaning

Before annotation and model training, the merged classification corpus underwent a preprocessing phase intended to improve consistency and reduce noise. Since the data

originated from different sources, variations in formatting, structure and writing style had to be addressed to create a more uniform dataset.

The preprocessing stage included the removal of duplicate or incomplete entries, normalization of the textual format and general cleaning operations such as lowercasing and the removal of irrelevant symbols or special characters. In some cases, requirement statements were also slightly standardized to improve consistency across the dataset and facilitate downstream processing by the classification models.

These preprocessing operations were considered important for improving the overall quality of the dataset and ensuring more reliable training and evaluation of the proposed models.

Figure 1 illustrates the workflow followed for dataset construction, including data collection, preprocessing, manual annotation, disagreement resolution and the creation of the final labelled dataset.

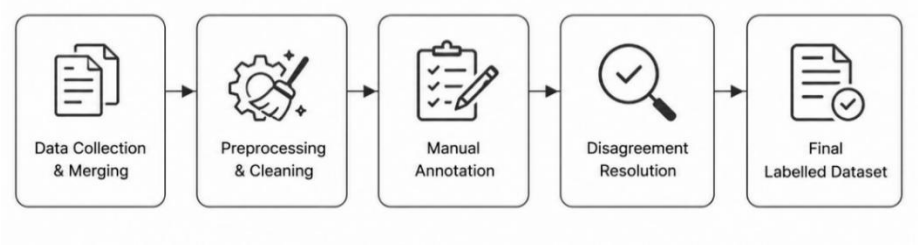

Figure 1: Dataset construction workflow

4.5 Annotation Procedure for Privacy Classification

Following preprocessing, the main corpus was manually annotated for the binary privacy classification task. Each requirement was assigned one of two labels privacy (1) or non-privacy (0). This binary labelling scheme was selected because the first stage of the system is concerned specifically with identifying whether a requirement belongs to the privacy domain, before any further legal or semantic analysis is performed.

The annotation process involved two independent annotators, both of whom reviewed the requirement statements and assigned labels. Cases involving clearly functional or unrelated software behaviour were labelled as non-privacy. The use of two annotators was intended to reduce subjectivity and improve the reliability of the resulting ground truth. One of the annotators is a bachelor student and the other annotator is a doctoral candidate working towards developers' privacy tools.

As expected, disagreements mainly emerged in cases where privacy aspects were implicit or ambiguously expressed, particularly in requirements related to general data-handling or security-related behaviour. Such ambiguities are common in privacy requirements analysis and have also been discussed in prior research on privacy requirement detection [2].

4.6 Handling Annotation Disagreements

To resolve annotation disagreements, a third reviewer examined the disputed cases and determined the final label for each requirement where the initial annotators differed. The third annotator is a senior researcher with experience on GDPR compliance. Out of the 2,614 annotated requirements, 469 cases involved disagreement, corresponding to a percentage disagreement rate of 17.94%.

Inter-annotator agreement was additionally evaluated using Cohen's Kappa coefficient [27], which measures the level of agreement between annotators while accounting for agreement occurring by chance. Cohen's Kappa is defined as:

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

where $P_o$represents the observed agreement between annotators and $P_e$represents the agreement expected by chance. The calculated Cohen's Kappa score for the annotation process was 0.5682, indicating moderate agreement between the annotators according to the interpretation guidelines proposed by Landis and Koch [27].

After reviewing the disagreements, it was observed that the third reviewer's judgments aligned more closely with those of the second annotator. On this basis, the labels of the second annotator were retained as the final labels for the disputed cases in most cases but not all. This adjudication process ensured consistency in the annotation decisions while preserving the overall reliability of the dataset.

The finalized labels were stored in a dedicated column and used as the ground truth for training and evaluation in the first stage of the proposed pipeline. This process contributes to the robustness of the dataset by reducing ambiguity and resolving inconsistencies introduced during independent annotation.

4.7 Manually Constructed Dataset for User Rights Classification

In addition to the main binary classification corpus, an extra manually constructed dataset was developed specifically for the second stage of the proposed system. This stage focuses on aligning privacy requirements with user rights and legal provisions. This dataset was not used for the first stage, instead, it was created to support the mapping privacy-related statements to specific legal concepts.

The creation of this additional dataset was necessary for two main reasons. First, the main classification corpus did not provide detailed annotations at the level of individual privacy rights or legal articles. Second, the available data did not adequately cover the full range of rights relevant to the thesis and present in the legislation. In other words, although the main corpus was suitable for training a classifier to distinguish privacy from non-privacy requirements, it was not balanced or complete for training a model to perform legal alignment at a finer granularity.

For this reason, a manual dataset was constructed to ensure that all target rights and article categories were represented. Initially, representative requirements were manually written for each target category to establish a foundational set of labelled examples. These examples were then used as seed instances in a prompt-based generation process. More specifically, the manually created requirements, together with their corresponding legal category, were provided to a LLM, which was instructed to generate additional software requirements semantically related to the same privacy right or legal concept.

Prompts structure:

“Given the following examples of requirements related to [CATEGORY], generate additional software requirements that preserve the same underlying privacy concept while varying the application context, wording, and requirement structure.”

The purpose of this process was not to replace manual judgment, but to extend the coverage of the dataset and provide additional training material, particularly in cases where naturally occurring examples were limited. Representative examples produced during this process included requirements such as: “The platform shall provide authenticated users with the ability to download a copy of their account information in

a machine-readable format", "Users shall be able to temporarily disable personalized recommendation features without affecting core system functionality", and "Inactive customer records shall be automatically removed after the expiration of the defined retention period". These examples demonstrate how the generated requirements varied in wording and application context while still representing the same legal concept.

This additional dataset was used exclusively for the training and support of the second-stage model. Its role was to compensate for the limited size and incomplete rights coverage of the primary corpus and to enable the system to learn clearer distinctions between closely related legal categories. The separation of datasets by task also strengthens the methodological clarity of the thesis, as it distinguishes between the requirements of binary privacy detection and those of detailed legal alignment.

4.8 Overview of the Final Dataset

Overall, the dataset construction process in this thesis followed a layered and purpose-driven design. Rather than relying on a single general-purpose dataset, the study employed a merged corpus for privacy classification and an additional manually constructed dataset. This design decision was driven by both practical and methodological considerations, including the scarcity of suitable publicly available datasets, the distinct requirements of each stage of the pipeline and the need for explicit coverage of legal privacy categories.

Furthermore, the use of multiple data sources and manual annotation procedures contributed to the diversity and representativeness of the dataset, allowing the models to capture both explicit and implicit expressions of privacy in natural language requirements. The incorporation of a dedicated dataset for legal alignment also ensured that less frequent or more specialized rights were adequately represented, improving the system's ability to perform fine-grained classification.

As a result, the final data preparation strategy supports the full architecture of the proposed system. This structured approach not only enhances the performance of the models but also strengthens the overall validity and applicability of the proposed methodology.

Chapter 5 – Experimental Methodology and System Design

5. Methodology

This chapter presents the experimental methodology and system design adopted for the legal alignment of privacy requirements from natural language software documentation. The proposed approach follows a multi-stage pipeline that combines NLP techniques, machine learning models, semantic similarity methods and rule-based mechanisms to transform unstructured textual requirements into structured privacy-aware and compliance-oriented outputs.

The design of the methodology is motivated by several challenges identified in the literature, including the ambiguity of natural language requirements, the presence of implicit and context-dependent privacy-related statements, as well as the complexity of aligning legal privacy concepts with software requirements. Prior studies have shown that identifying privacy requirements from textual artifacts is inherently difficult due to linguistic variability, semantic ambiguity and the complexity of distinguishing privacy-related content from other requirements [6][13]. Furthermore, legal frameworks such as the GDPR and the CCPA introduce additional complexity because legal rights and obligations are often expressed indirectly.

To address these challenges, this thesis adopts a hybrid and experimentally driven methodology. Instead of relying on a single modelling paradigm, multiple approaches were explored throughout the development process, including traditional machine learning models, transformer-based architectures, semantic similarity techniques and prompt-based LLM. These experiments were used to identify the strengths and limitations of each method and guide the design of the final proposed system.

The overall system is organised into two main analytical stages. The first stage focuses on binary privacy classification and determines whether a requirement is privacy-related or not. This stage acts as a filtering mechanism that reduces irrelevant input and ensures that only potentially privacy-relevant requirements proceed to subsequent processing. The second stage performs legal-semantic analysis by mapping the identified privacy requirements to user rights and legal concepts derived from privacy regulations, specifically GDPR and CCPA.

Before these stages, a preprocessing phase is applied to normalize and prepare the textual input. This phase includes operations such as normalization, noise removal, formatting standardization and lightweight text cleaning. The objective of preprocessing is not to remove semantic information, but rather to improve consistency and robustness across classification and semantic matching tasks.

The resulting methodology therefore constitutes an end-to-end privacy analysis pipeline, beginning with raw textual requirements, continuing with privacy detection and concluding with legal rights mapping. In addition to automated analysis, the methodology also incorporates explainability and user validation mechanisms, enabling greater transparency during the decision-making process.

5.1 Overall Research Pipeline

The proposed system is designed as a structured multi-stage pipeline. The objective of the pipeline is to transform natural language requirements into structured representations aligned with legal concepts derived from privacy regulations.

The system follows a sequential processing workflow consisting of preprocessing, privacy classification, legal-semantic alignment and structured result generation.

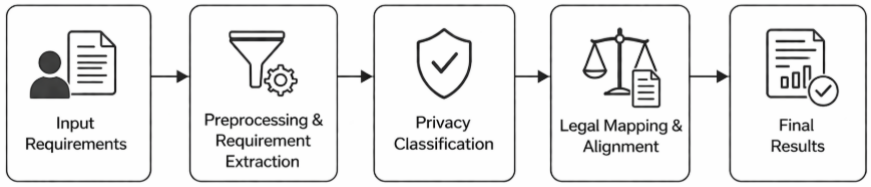

Figure 2: High-level overview of the proposed pipeline for privacy requirement detection and legal alignment.

As illustrated in the figure above, the pipeline begins with the input requirements, which may be provided either manually through textual input or through uploaded structured files. These requirements are first processed through a preprocessing phase that performs text normalization and formatting standardization.

Following preprocessing, the requirements are forwarded to the first stage, where the system determines whether each requirement is privacy-related or non-privacy-related. This stage is formulated as a binary classification task and acts as a filtering mechanism that prevents irrelevant system functionality from reaching the final stage.

Requirements identified as privacy-related are forwarded to the next stage of the pipeline, where they are semantically analysed and mapped to legal concepts and user rights derived from GDPR and CCPA. This stage attempts to capture both explicit and implicit legal meaning through semantic representations and domain-specific refinement mechanisms.

Finally, the pipeline produces structured outputs containing the predicted legal concepts.

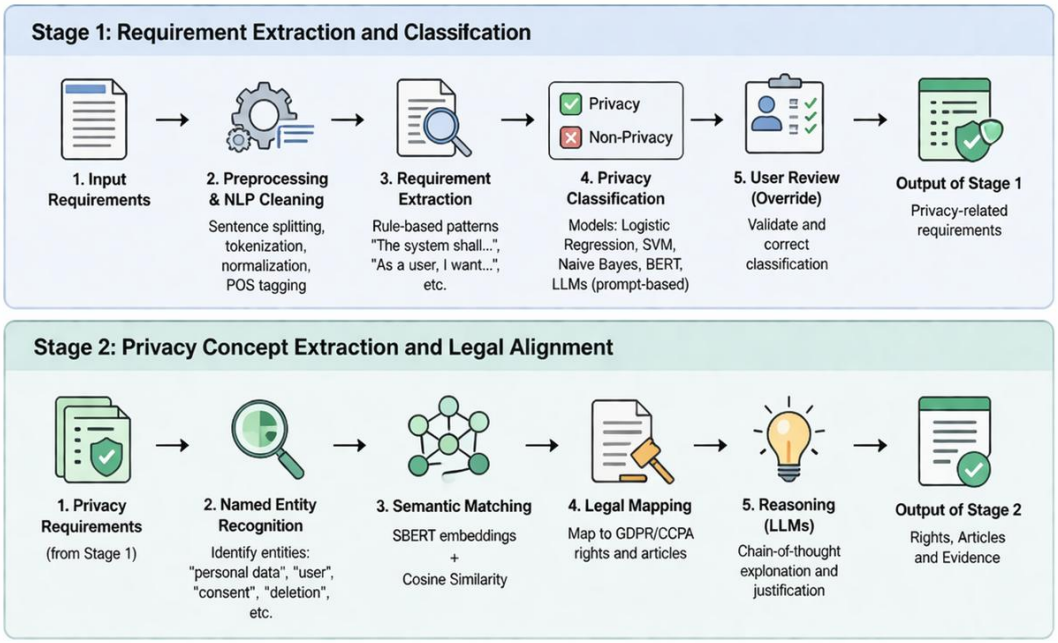

Figure 3: Detailed architecture of the proposed pipeline, illustrating the interaction between preprocessing, classification, semantic analysis, and legal alignment components.

The detailed pipeline architecture illustrates the interaction between machine learning models, semantic similarity modules, rule-based refinement mechanisms and explainability components.

The proposed pipeline bridges the gap between natural language software specifications and formal privacy regulations by combining automated analysis with structured outputs.

5.2 Dataset Preparation and Experimental Setup

The experimental methodology adopted in this thesis relies on multiple datasets and evaluation settings, reflecting the different objectives and requirements of the two stages of the proposed pipeline.

For Stage 1, the experiments are based on a corpus of annotated software requirements labelled as privacy-related (1) or non-privacy-related (0). The dataset consists of natural language requirements collected from publicly available datasets.

Different evaluation strategies are employed depending on the experimental setting and model type. Classical machine learning approaches are primarily evaluated using k-fold cross-validation to reduce sampling bias and improve robustness of the reported results. In contrast, transformer-based architectures and the final pipeline rely on separate validation subsets during training and model development, while maintaining independent test sets for final evaluation.

The preprocessing pipeline includes text normalization, whitespace normalization and lightweight text cleaning. Different preprocessing strategies are applied depending on the model type. Traditional machine learning models use more normalized textual representations, while transformer-based architectures preserve richer contextual information.

For traditional machine learning approaches, preprocessing primarily focuses on reducing lexical variability and improving feature consistency across requirements. This includes normalization operations such as lowercasing, whitespace normalization and textual cleaning. These steps are particularly beneficial for TF-IDF-based lexical representations [24], since they help improve feature consistency across textual requirements.

In contrast, transformer-based architectures rely more heavily on contextual language modelling and bidirectional attention mechanisms, allowing them to better utilize contextual semantic and linguistic information [15][17]. Since transformer models internally learn contextual relationships between words and surrounding text, aggressive normalization and extensive manual feature engineering are generally minimized to avoid removing potentially meaningful contextual signals. This distinction reflects the different representational assumptions underlying classical lexical models and contextual embedding architectures.

The second stage of the pipeline introduces additional complexity, since requirements may correspond to multiple legal concepts simultaneously. To support this task, an additional manually constructed dataset is introduced.

The manually constructed dataset was specifically designed to enrich the representation of privacy rights and legal concepts that were underpresented in the original data. These manually created requirements include implicit privacy requirements, semantically ambiguous requirements, edge cases involving overlapping legal rights, indirect references to privacy concepts and complex multi-label scenarios. This enrichment process improves the diversity and representativeness of the legal mapping dataset and enables more robust evaluation of semantic capabilities.

For the final mapping architecture, the main dataset was divided into training, validation and test subsets using a three-way split strategy. Specifically, 30% of the main dataset was reserved for final testing, while the remaining portion was further divided to create a dedicated validation subset used during model development. The manually constructed dataset was incorporated only into the training pool to improve label coverage and support the learning of less frequent legal categories.

The use of a three-way split strategy allowed the experimental process to preserve an independent test set for final evaluation while still enabling intermediate evaluation of the architecture during development. In this way, the final test results provide a more reliable estimate of the model's generalization performance on previously unseen requirements.

The final label space integrates legal categories derived from GDPR and CCPA.

Regulation	***Article***	***Description***
GDPR	Article 5	Principles of data processing (lawfulness, fairness, transparency, minimization)
GDPR	Article 6	Lawfulness of processing
GDPR	Article 7	Conditions for consent
GDPR	Articles 12–14	Transparency and information obligations
GDPR	Article 15	Right of access to personal data
GDPR	Article 16	Right to rectification of inaccurate data
GDPR	Article 17	Right to erasure
GDPR	Article 18	Right to restriction of processing
GDPR	Article 20	Right to data portability
GDPR	Article 21	Right to object to processing
GDPR	Article 22	Automated decision-making and profiling
GDPR	Article 25	Data protection by design and by default
GDPR	Article 29	Processing under authority

GDPR	Article 30	Records of processing activities
GDPR	Articles 31–34	Security, breach notification, and cooperation
CCPA	Right to Know	Users can know what personal data is collected
CCPA	Right to Access	Users can access their personal data
CCPA	Right to Delete	Users can request deletion of their personal data
CCPA	Right to Correct	Users can correct inaccurate personal data
CCPA	Right to Opt-Out	Users can opt out of data sale or sharing
CCPA	Right to Non-Discrimination	Protection from discrimination when exercising rights

Table 3: Legal categories used as labels in Stage 2, covering GDPR articles and CCPA user rights.

Overall, the dataset preparation process supports both the methodological exploration and the final system design, ensuring that the proposed pipeline is evaluated under realistic and semantically diverse conditions.

5.3 Privacy Classification Approaches

The first stage of the proposed pipeline focuses on the automatic classification of requirements into privacy-related and non-privacy-related categories. This task is formulated as a supervised binary classification problem.

The primary objective of this stage is to act as a filtering mechanism. Only requirements identified as privacy-related are forwarded to the next stage. Consequently, minimizing false negatives is particularly important, as incorrectly discarding a privacy requirement would prevent further legal analysis and regulatory mapping.

5.3.1 Classical Machine Learning

As an initial step, several traditional machine learning approaches were explored, including Support Vector Machines (SVM), Logistic Regression and Naive Bayes.

These models are widely used in text classification tasks due to their computational efficiency and strong performance on sparse textual representations.

The requirements were represented using TF-IDF vectorization, incorporating both word-level and character-level n-grams. Word-level representations capture lexical semantics, while character-level features improve robustness against linguistic variation and spelling differences.

The objective of these baseline experiments was to establish a strong reference point before transitioning to more advanced contextual models.

5.3.2 Transformer-Based Classification

To address the limitations of traditional models in capturing contextual and implicit privacy semantics, transformer-based approaches were explored.

The transformer architecture selected for Stage 1 is based on BERT (Bidirectional Encoder Representations from Transformers) [15]. BERT is particularly suitable for this task because it captures bidirectional contextual information through transformer self-attention mechanisms, allowing the model to jointly interpret both preceding and succeeding textual context during semantic analysis. This capability is especially important for privacy-related requirements, where legal or privacy implications are often expressed implicitly rather than through explicit privacy keywords alone.

During preprocessing, each requirement is tokenized using the BERT tokenizer and transformed into contextual embeddings. The contextual representation associated with the special [CLS] token is subsequently used as the semantic representation of the requirement and provided as input to a final binary classification layer that predicts whether the requirement is privacy-related or non-privacy-related.

Compared to classical lexical approaches, BERT enables improved understanding of semantically complex and context-dependent privacy language.

Since the dataset contains a substantially smaller number of privacy-related requirements compared to non-privacy requirements, weighted cross-entropy loss is applied during training to reduce the impact of class imbalance and improve sensitivity toward the minority privacy class [20].

5.3.3 Threshold Analysis and Rule-Based Refinement

In addition to probabilistic classification, threshold analysis is incorporated into the experimental methodology. Rather than relying on a fixed classification threshold, multiple threshold values are explored to analyse the trade-off between precision and recall. This is very important in privacy detection tasks, where strict thresholds may lead to missed privacy-related requirements.

The transformer-based classifier is further complemented by a rule-based refinement layer. This refinement mechanism incorporates domain-specific patterns associated with personal data, consent, deletion, rectification, access rights, opt-out mechanisms, data sharing, retention policies and breach-related terminology. At the same time,

separate rule groups are used to suppress false positives related to purely technical or system-level functionality. This hybrid decision strategy combines the contextual capabilities of BERT with deterministic domain-informed corrections.

5.4 Legal Alignment of Privacy Requirements.

The second stage of the pipeline focuses on analysing privacy requirements and associating them with relevant legal rights and regulatory concepts.

Unlike Stage 1, which performs binary classification, this stage addresses a substantially more complex problem involving semantic ambiguity, overlapping legal concepts and implicit privacy meaning.

5.4.1 The Challenge of Multi-Label Legal Alignment

The task addressed in Stage 2 is formulated as a multi-label classification problem.

A single requirement may correspond to multiple legal rights simultaneously. For example, a requirement allowing users to access and download their data may correspond both to the right of access and the right to data portability.

Formally, the objective is to learn a mapping between natural language requirements and subsets of legal labels derived from GDPR and CCPA.

The label taxonomy includes GDPR Articles 5, 6, 7, 12–22, 25 and 29–34, together with CCPA rights such as the right to know, right to access, right to delete, right to correct and right to opt out

The complexity of the task arises from semantic overlap between legal rights, implicit privacy expressions, ambiguity in natural language and the absence of explicit legal terminology in many requirements.

Unlike traditional single-label classification tasks, this mapping problem requires the model to identify multiple potentially overlapping legal concepts within the same requirement. In many cases, semantically similar requirements may correspond to different legal provisions, while multiple rights may apply to a single functionality. Furthermore, several legal concepts share partially overlapping terminology, making strict boundary separation difficult. These characteristics introduce substantial semantic complexity and motivate the use of hybrid lexical, semantic and rule-based modelling strategies.

5.4.2 Rule-Based and Feature-Based Techniques

As an initial approach, rule-based pattern matching mechanisms were explored to capture explicit legal and privacy-related terminology within software requirements.

These approaches relied on manually defined lexical patterns and regular expressions associated with specific legal rights and regulatory concepts. For example, deletion-related expressions were associated with erasure rights, while export- and download-related expressions were linked to portability and access-related rights. Additional patterns were also introduced for concepts such as consent management, opt-out mechanisms, transparency obligations and security-related provisions.

Although rule-based methods demonstrated strong precision for explicit privacy statements, they struggled with implicit, context-dependent and semantically ambiguous cases. Requirements expressing privacy functionality indirectly, or using highly variable linguistic formulations, frequently could not be captured reliably through manually defined lexical rules alone.

5.4.3 Multiclass Transformer-Based Classification

A transformer-based multiclass classification was also explored as an alternative approach for legal rights prediction.

In this setup, each requirement was mapped directly to a legal label using DistilBERT, a computationally efficient transformer-based language model derived from BERT and designed to preserve much of BERT's contextual language understanding capabilities while reducing computational complexity [16]. The model was fine-tuned on privacy-related requirements associated with GDPR and CCPA labels.

The input requirements were tokenized using the DistilBERT tokenizer and converted into contextual embeddings. The contextual representation associated with the special classification token was then used through a softmax classification layer to predict the final legal category.

Since the dataset exhibited class imbalance, weighted cross-entropy loss was employed during training to reduce bias toward dominant legal categories. Class weights were computed inversely proportional to class frequencies in the training data, assigning larger penalty values to underrepresented labels and smaller weights to highly frequent

categories. This weighting strategy encouraged the model to pay greater attention to rare legal classes during optimization.

Several training configurations were experimentally evaluated, including different learning rates, batch sizes, weight decay values and training epochs, in order to improve generalization and reduce overfitting.

The multiclass formulation was investigated to evaluate whether a transformer-based approach could model contextual relationships between requirements and legal concepts more effectively than traditional feature-engineering methods.

5.4.4 Semantic Pairwise Classification with BERT

An alternative formulation based on pairwise semantic verification was also investigated. In this approach, the legal mapping task was modelled as a semantic matching problem instead of direct label classification. Rather than predicting a legal category directly from the requirement text, the model evaluated whether a requirement semantically corresponded to a candidate legal article or privacy right. Each pair was treated as either a valid or non-valid semantic match.

In this formulation, the input to the model consisted of requirement–article pairs. Each pair included a natural language software requirement together with a GDPR article or CCPA right. The objective of the model was to determine whether the requirement semantically corresponded to the associated legal concept.

This approach is conceptually related to semantic textual similarity methods and sentence-pair modelling approaches commonly used in information retrieval and semantic similarity tasks [17]. The formulation was motivated by the observation that legal concepts often share overlapping terminology and semantic relationships, making direct multiclass prediction particularly difficult.

To construct the training dataset, positive and negative pairs were generated. Positive pairs (1) corresponded to valid legal mappings derived from annotated data, while negative pairs (0) were created by associating requirements with unrelated legal labels. Since naive pair generation produced substantially more negative than positive examples, controlled negative sampling and balancing strategies were incorporated during dataset construction to improve semantic discrimination, stabilize training and reduce prediction bias toward negative predictions.

Pairs were encoded using the DistilBERT transformer model [16], allowing contextual representations to be learned jointly for both the requirement text and the candidate legal concept. The task was formulated as a binary semantic verification problem, where the model predicted whether a requirement and legal label pair represented a valid legal correspondence.

The pairwise formulation was explored to evaluate whether semantic verification between requirements and candidate legal concepts could improve the identification of subtle or context-dependent legal relationships compared to direct classification approaches.

5.4.5 Prompt-Based LLM Exploration

A prompt-based large language model approach was explored as an additional experimental direction for the legal mapping task. Unlike supervised machine learning methods, these approaches relied on instruction-driven reasoning capabilities without performing task-specific model fine-tuning.

The Meta Llama-3.2-3B-Instruct [18] model was selected due to its strong instruction-following and zero-shot reasoning capabilities, which make it appropriate for exploring semantic interpretation and legal mapping tasks without supervised fine-tuning.

The experiments employed Meta Llama-3.2-3B-Instruct [18] in a zero-shot setting, where the model received natural language requirements together with textual descriptions of legal rights and was asked to predict the most appropriate legal mapping. This formulation was investigated to evaluate whether large language models could generalize to legal tasks through semantic reasoning alone.

Multiple prompting strategies were examined throughout the experiments. Initial prompts relied on zero-shot prompting strategies, where the model was instructed to directly predict legal labels from requirement text without task-specific training examples [19]. Subsequently, more structured prompts were explored to improve output consistency, reasoning quality and semantic alignment.

These structured prompts incorporated semantic descriptions of GDPR and CCPA rights, explicit formatting constraints, reasoning-oriented decomposition steps and predefined decision rules intended to guide the model towards more consistent legal mapping. This design was motivated by prior research demonstrating that chain-of-

thought prompting can improve reasoning performance by guiding models through intermediate reasoning steps [25].

Additional constraints were introduced in certain prompt variants to improve output consistency and enforce valid label generation, consistent with prompt engineering research emphasizing the importance of carefully designed prompts for guiding and controlling model behaviour [19].

The experiments also investigated the impact of prompt complexity and instruction specificity on prediction stability and semantic consistency. Attention was given to the ability of the model to distinguish semantically related legal concepts and identify implicit privacy requirements.

Unlike the supervised approaches explored in previous experiments, the prompt-based formulation did not rely on gradient-based training or dataset-specific optimization. Instead, the methodology focused on evaluating the zero-shot reasoning and instruction-following capabilities of instruction-tuned LLMs within the context of privacy requirement analysis.

The overall objective of these experiments was to assess whether instruction-driven LLMs could directly perform legal-semantic mapping without requiring dedicated supervised training pipelines.

5.4.6 Proposed Hybrid Pipeline

The findings obtained from the experimental exploration ultimately motivated the design of a hybrid architecture.

Step	*Technique / Model*	*Role in Pipeline*	*Purpose*
1	Word-level TF-IDF	Lexical feature extraction	Capture explicit privacy terminology and legal expressions appearing in software requirements
2	Character-level TF-IDF	Subword feature extraction	Capture morphological variations and subword linguistic patterns related to privacy concepts
3	Sentence-BERT embeddings	Semantic representation	Capture contextual semantic meaning and identify implicit privacy-related information beyond keyword similarity

4	Prototype similarity matching	Semantic similarity estimation	Compare requirement embeddings with manually defined GDPR and CCPA concept descriptions
5	One-vs-Rest Logistic Regression	Multi-label classification	Train one binary classifier for each legal label and estimate whether each GDPR article or CCPA right applies to the requirement.
6	Dynamic label selection	Adaptive prediction refinement	Select candidate legal labels based on prediction confidence and semantic relevance
7	Deterministic cross-regulatory coupling	Cross-framework consistency enforcement	Align semantically equivalent GDPR and CCPA rights during post-processing
8	Rule-based refinement mechanisms	Final prediction adjustment	Improve semantic consistency and reduce incorrect or ambiguous legal mappings

Table 4: Steps of the Proposed Hybrid Pipeline

The final Stage 2 architecture combines lexical, semantic and rule-based components to support the legal alignment of privacy requirements. Table 4 summarizes the main steps and components of the proposed hybrid pipeline. Lexical TF-IDF representations at both word and character level are used to capture explicit privacy-related expressions and subword linguistic patterns [23], while Sentence-BERT embeddings provide contextual semantic representations capable of identifying implicit privacy meaning beyond surface-level keyword similarity [17].

The One-vs-Rest classification strategy decomposes the multi-label prediction problem into multiple independent binary classification tasks, where a separate classifier is trained for each legal label. This allows the system to independently estimate the relevance of each GDPR article or CCPA right for a given requirement, making the approach particularly suitable for privacy analysis scenarios in which multiple legal rights may simultaneously apply. Logistic Regression was selected as the final prediction layer due to its strong performance in high-dimensional textual classification tasks and computational efficiency. Furthermore, the architecture combines heterogeneous feature representations, including lexical TF-IDF features, semantic embeddings and prototype similarity scores, within a unified prediction framework.

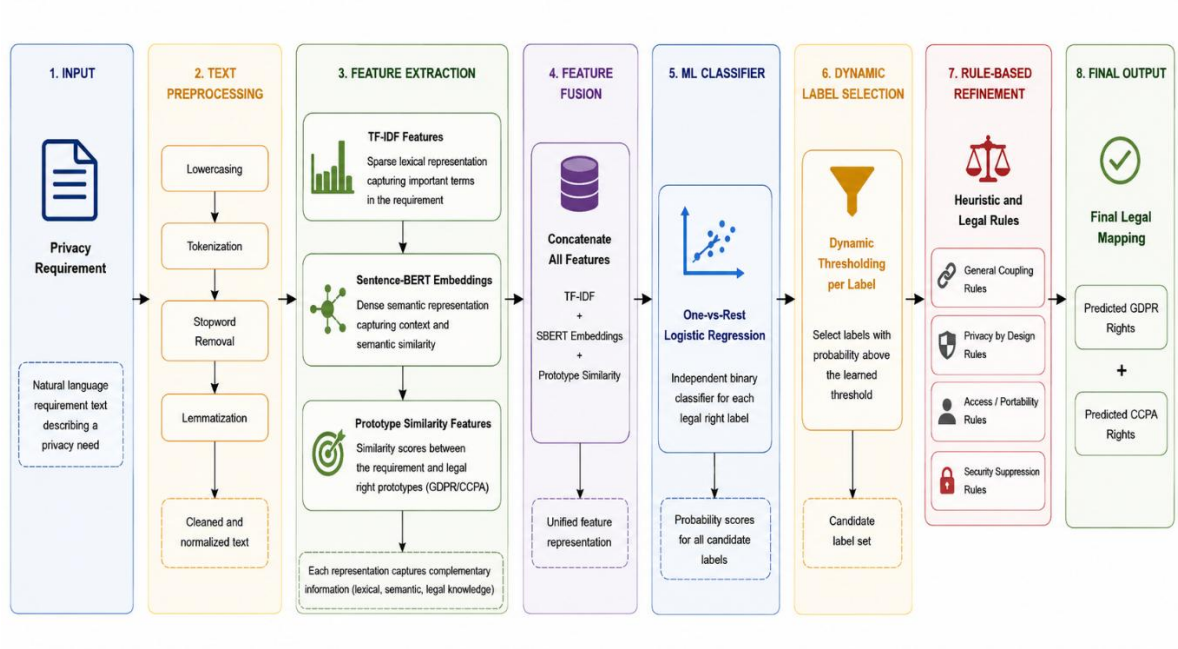

Figure 4: Proposed Hybrid Architecture for Legal-Semantic Mapping of Privacy Requirements

Figure 4 illustrates the overall structure and interaction of the components comprising the final hybrid mapping architecture.

One of the core components of the system is the prototype similarity mechanism. In this approach, each legal label is associated with a semantic prototype description representing the meaning of the corresponding GDPR article or CCPA right. These prototype descriptions are manually constructed semantic summaries of the corresponding legal concepts, allowing the model to estimate semantic proximity between requirements and legal rights during classification.

In addition to semantic similarity features, the prediction process also incorporates adaptive label selection. Instead of relying on a single fixed threshold for all labels, the system dynamically selects candidate labels based on confidence scores and semantic proximity. The highest-scoring label is always retained, while additional labels are included only when their scores satisfy predefined selection conditions.

To ensure consistency across regulatory frameworks, the system incorporates a deterministic label coupling mechanism during post-processing. Rather than treating semantically aligned GDPR and CCPA rights as entirely independent outputs, the model enforces selected cross-regulatory pairings in cases where strong functional equivalence exists. These pairings were selected based on strong functional correspondences between the two regulatory frameworks, as also reflected in comparative analyses of GDPR and CCPA/CPRA rights [22]. While broader conceptual

relationships may be identified in the literature, only this subset is explicitly encoded in the current implementation.

GDPR Label	***CCPA Label***
Article 15 – Right of Access	Right to Access
Article 16 – Right to Rectification	Right to Correct
Article 17 – Right to Erasure	Right to Delete
Article 21 – Right to Object	Right to Opt-Out of Sale or Sharing

Table 5: Cross-regulatory label couplings explicitly implemented in the Stage 2 post-processing layer.

Overall, the coupling mechanism improves semantic consistency between equivalent GDPR and CCPA concepts while preserving the flexibility of the multi-label prediction process.

5.5 Overall System Workflow

The proposed methodology is implemented as an end-to-end privacy analysis workflow integrating automated inference, semantic analysis, explainability and user validation. Figure 4 presents the overall operational workflow of the proposed system, beginning from requirement submission through either manual textual input or file upload. The uploaded content is first pre-processed and analysed in Stage 1, where the system identifies candidate privacy-related requirements through automated classification mechanisms.

The intermediate results are subsequently presented to the user for validation before proceeding to legal analysis, introducing a human-in-the-loop workflow that improves robustness and allows users to revise ambiguous or potentially incorrect classifications. Once a requirement is confirmed as privacy-related, Stage 2 of the pipeline is activated. During this stage, the system performs legal-semantic mapping between the identified requirement and the corresponding GDPR or CCPA rights.

Before the final mapping process, the workflow additionally evaluates whether the requirement contains multiple coordinated clauses or semantically distinct actions. Complex requirements may therefore be decomposed into smaller sub-requirements, which are analysed independently to improve semantic precision and reduce ambiguity during the legal mapping process.

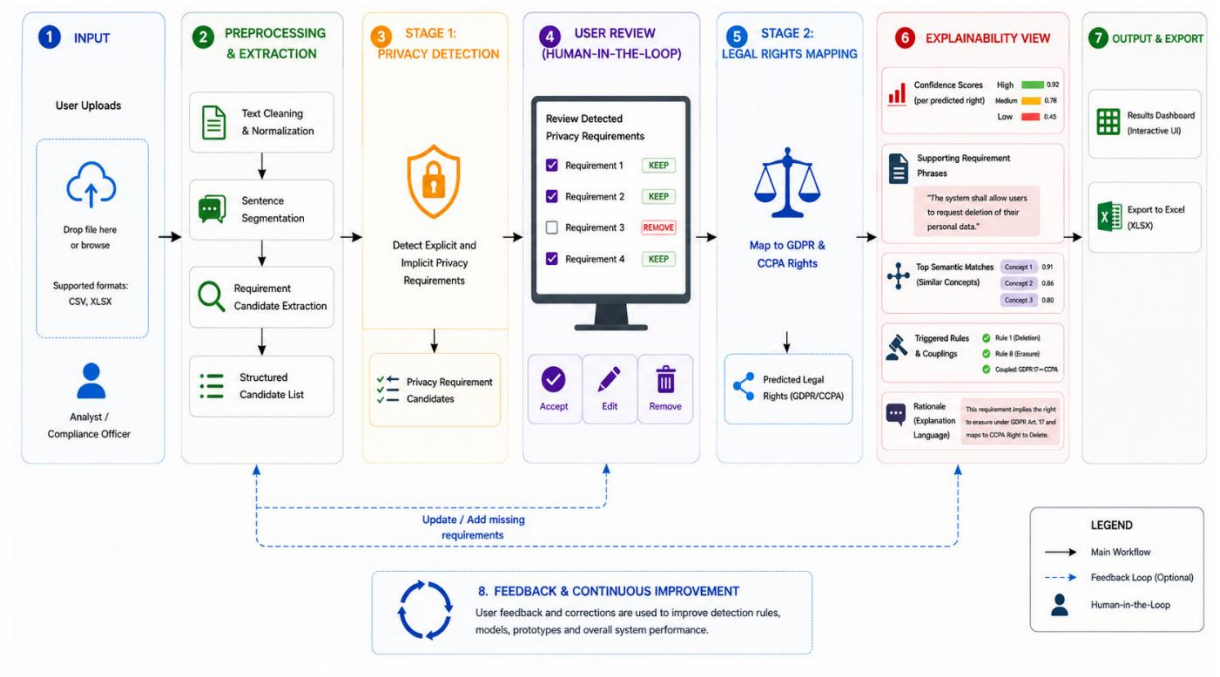

Figure 5: End-to-End Workflow and Explainability Process of the Proposed System

In summary, the proposed workflow combines automated machine learning techniques with controlled human interaction and semantic analysis,

Chapter 6 – Evaluation and Results

This chapter presents the evaluation of the proposed pipeline across the stages of the system. The objective of the evaluation is to assess the ability of the proposed approach to identify privacy-related software requirements and subsequently align them with legal rights and regulatory concepts derived from GDPR and CCPA.

The evaluation examines the performance of the proposed approaches, their prediction behaviour and the types of errors produced during legal mapping. Emphasis is placed on challenges introduced by implicit privacy semantics, semantic overlap between legal concepts and the multi-label nature of legal rights mapping.

6.1 Experimental Configuration

This section outlines the evaluation protocol followed in this chapter. Since the proposed system consists of two stages, the evaluation is also organized in two parts, privacy classification and legal rights mapping. The first stage is evaluated as a binary classification task, while the second stage is evaluated as a multi-label classification and ranking problem.

For Stage 1, the evaluation focuses on the ability of the system to distinguish privacy-related from non-privacy-related requirements. The main metrics used are accuracy, precision, recall, F1-score and macro-F1. Particular importance is given to recall for the privacy class, as incorrectly rejecting privacy-related requirements would exclude them from subsequent legal analysis.

For Stage 2, the evaluation focuses on the ability of the system to associate privacy-related requirements with relevant GDPR and CCPA concepts. Since each requirement may correspond to more than one legal right, the evaluation uses micro-averaged, macro-averaged and sample-based precision, recall and F1-score. In addition, ranking-based evaluation metrics were used to assess whether the correct legal labels appeared among the highest-ranked predictions.

The evaluation metrics used for the assessment of both stages of the proposed pipeline are summarized in Table 6.

Stage	*Evaluation Aspect*	*Evaluation Metrics*	*Evaluation Objective*
Stage 1	Binary Privacy Classification	Accuracy, Precision, Recall, F1-score	Evaluate overall classification performance
Stage 2	Multi-label Legal Rights Mapping	Micro-Precision, Micro-Recall, Micro-F1, Macro-Precision, Macro-Recall, Macro-F1, Sample-F1	Evaluate multi-label prediction performance
Stage 2	Ranking-based Evaluation	Recall, Coverage	Evaluate top-ranked predictions

Table 6: Evaluation metrics used across both stages of the proposed pipeline.

6.2 Privacy Classification Results

The Stage 1 dataset exhibits a noticeable imbalance between privacy and non-privacy requirements, reflecting realistic software engineering environments where privacy-related requirements typically constitute only a subset of overall system functionality.

The class imbalance affects model behaviour because models may become biased toward the dominant non-privacy class. As a result, privacy-related requirements may be incorrectly classified as non-privacy-related. Since Stage 1 determines which requirements proceed to legal analysis, reducing such false negatives is very important.

6.2.2 Comparative Results of Classical and BERT-Based Models

Multiple classification approaches were evaluated during the experiments, including Support Vector Machines (SVM), Logistic Regression, Naive Bayes and BERT.

Model	*Accuracy*	*Precision*	*Recall*	*F1*	*Macro-F1*
SVM	0.817	0.766	0.720	0.742	0.800
LR	0.815	0.761	0.720	0.740	0.798
Naive Bayes	0.752	0.659	0.667	0.663	0.733
BERT	0.780	0.668	0.792	0.725	0.771

Table 7: Performance comparison of classification models on the privacy detection task

The results show that SVM and Logistic Regression achieved the strongest performance among the classical machine learning approaches. SVM achieved the highest accuracy (0.817) and macro-F1 score (0.800), while Logistic Regression produced similar results across all evaluation metrics. These findings suggest that TF-

IDF representations combined with linear classifiers are highly effective for detecting explicitly expressed privacy-related terminology and lexical privacy patterns.

In contrast, Naive Bayes produced noticeably lower performance, achieving the lowest F1-score (0.663) and macro-F1 score (0.733). This behaviour indicates limitations in contextual dependencies and semantic variability within natural language requirements, particularly when privacy meaning depends on sentence structure rather than isolated keywords.

Although the BERT model achieved slightly lower overall accuracy (0.780) compared to SVM, it achieved the highest recall for the privacy class (0.792). This result is particularly important within the context of the proposed pipeline because false negatives directly reduce compliance coverage by preventing privacy-related requirements from reaching Stage 2 legal analysis.

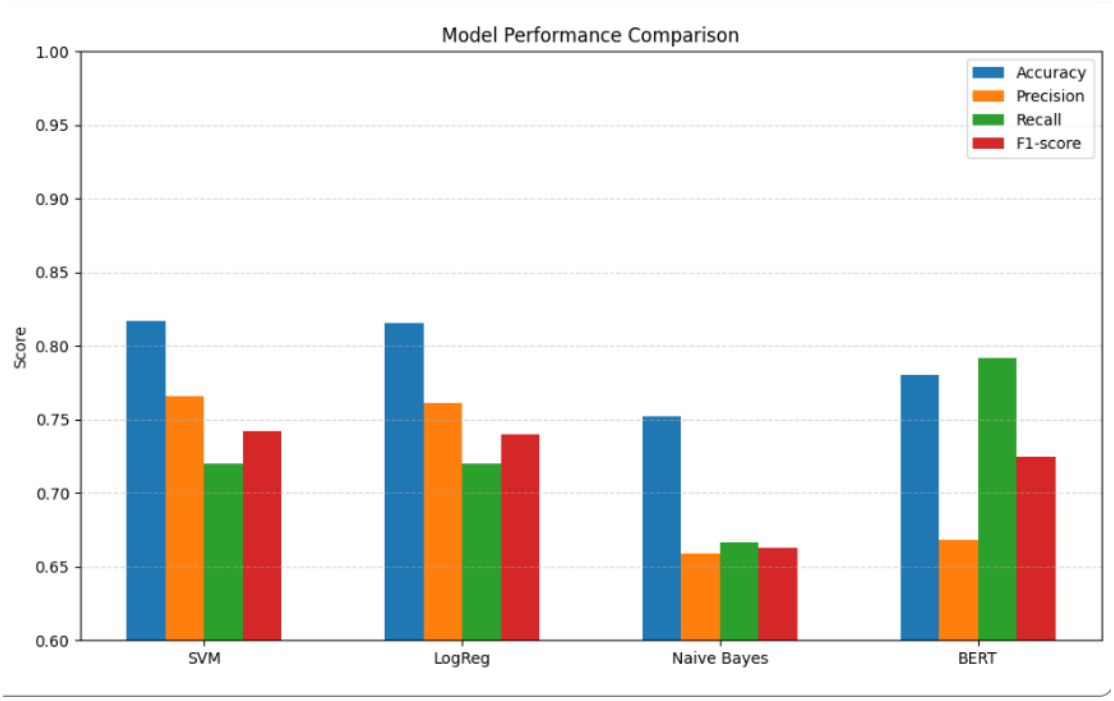

Figure 6: Comparison of model performance across evaluation metrics

The higher recall achieved by BERT suggests that contextual transformer representations improve the detection of implicit privacy semantics. In particular, the model demonstrated improved sensitivity to requirements involving tracking, personalization, indirect monitoring and contextual data usage even when explicit privacy-related terminology was absent.

These findings indicate that contextual language understanding plays an important role in identifying implicit privacy requirements that are difficult to capture using purely lexical approaches.

6.2.3 Classification Errors and Confusion Analysis

To further analyse classifier behaviour, confusion matrix analysis was performed for both SVM and BERT models.

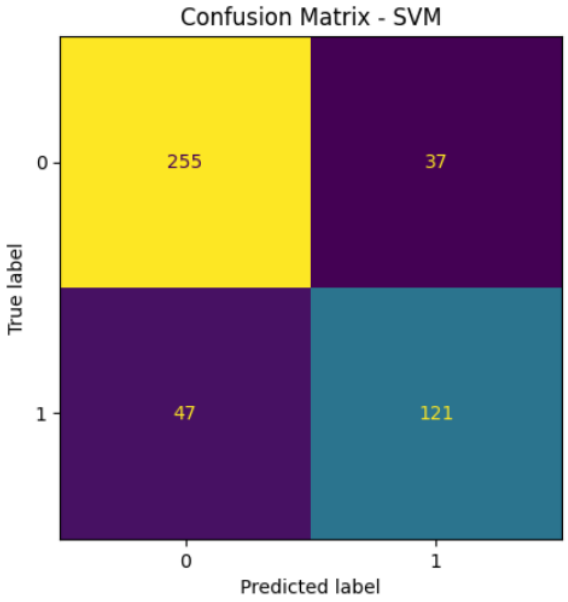

Figure 7: Confusion matrix of the SVM model on the test set

The SVM classifier demonstrated strong capability in identifying non-privacy requirements, producing a high number of true negatives (255) while maintaining relatively balanced precision and recall. However, the model also produced 47 false negatives, indicating that several privacy-related requirements were incorrectly classified as non-privacy-related.

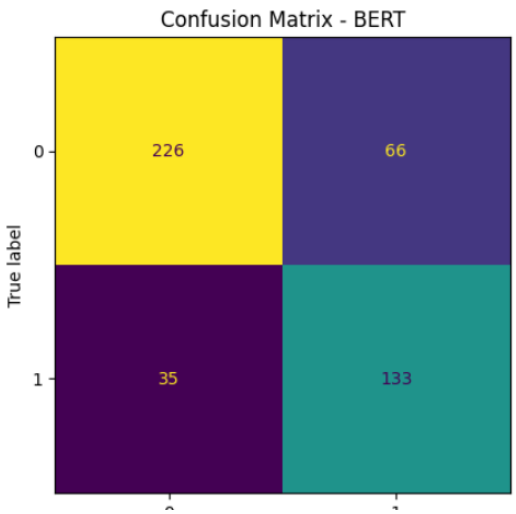

Figure 8: Confusion matrix of the BERT model on the test set

The BERT confusion matrix further highlights its recall-oriented behaviour. Compared to SVM, BERT reduced the number of false negatives to 35 while increasing the number of true positives to 133. These results confirm the improved sensitivity of contextual transformer representations to implicit privacy semantics.

However, the improved recall was accompanied by a noticeable increase in false positives. BERT produced 66 false positives compared to 37 for SVM, indicating that the model occasionally classified security-oriented or technical functionality as privacy-related.

Several false positives corresponded to requirements describing security-related system functionality, including user authentication mechanisms, monitoring operations and protected access to sensitive information. These findings demonstrate the strong semantic overlap between privacy and security concepts in natural language software requirements. In many cases, requirements contain terminology associated with both domains, making strict separation inherently difficult even for contextual models, a challenge also identified in prior privacy requirements engineering research [26].

The error analysis additionally highlights the importance of balancing recall and precision. Highly conservative models increase false negatives and risk excluding privacy-related requirements from legal analysis, while highly recall-oriented models introduce additional non-relevant requirements into downstream processing.

Overall, the evaluation results indicate that transformer-based contextual representations provide important advantages for privacy requirement detection. Although SVM achieved slightly higher overall accuracy and macro-F1 performance,

the improved recall and contextual sensitivity demonstrated by BERT make it more suitable for downstream legal analysis within the proposed pipeline.

6.3 Legal Alignment Results

6.3.1 Challenges of Multi-Label Legal Alignment

The legal rights mapping task proved substantially more challenging than binary privacy classification due to the semantic complexity of legal concepts and the multi-label nature of the problem. Several factors contributed to this increased difficulty, including semantic overlap between legal rights, implicit legal meaning and label imbalance.

In many cases, a single requirement corresponded to multiple legal concepts. For example, requirements involving user data export functionality frequently overlapped between access and portability rights. Similarly, authentication and monitoring requirements frequently exhibited semantic overlap between privacy-oriented and security-oriented legal provisions.

Another important challenge involved the implicit expression of legal concepts. Many requirements described privacy-related behaviour indirectly through functional system descriptions rather than explicit regulatory terminology. As a result, successful legal mapping required contextual semantic analysis rather than simple keyword matching.

These findings demonstrate that legal rights mapping constitutes a more complex problem than traditional classification tasks.

6.3.2 Analysis of Experimental Approaches

Multiple experimental approaches were evaluated to investigate different modelling strategies for legal rights mapping and semantic alignment. The following sections present the results obtained from each experimental configuration.

6.3.2.1 Multiclass Transformer-Based Classification Results

The first experimental approach reformulated the problem as a multiclass classification task using DistilBERT.

Statistic	*Value*
Total number of requirements	346
Number of classes	15
Training set size	276
Validation set size	70
Minimum samples per class	5
Maximum samples per class	94
Average samples per class	23.1

Table 8: Summary of Dataset Statistics After Preprocessing and Filtering

The filtered dataset used for this experiment contained 346 privacy requirements distributed across 15 legal classes. However, the distribution of samples across classes remained highly imbalanced, with several labels containing only a limited number of training examples.

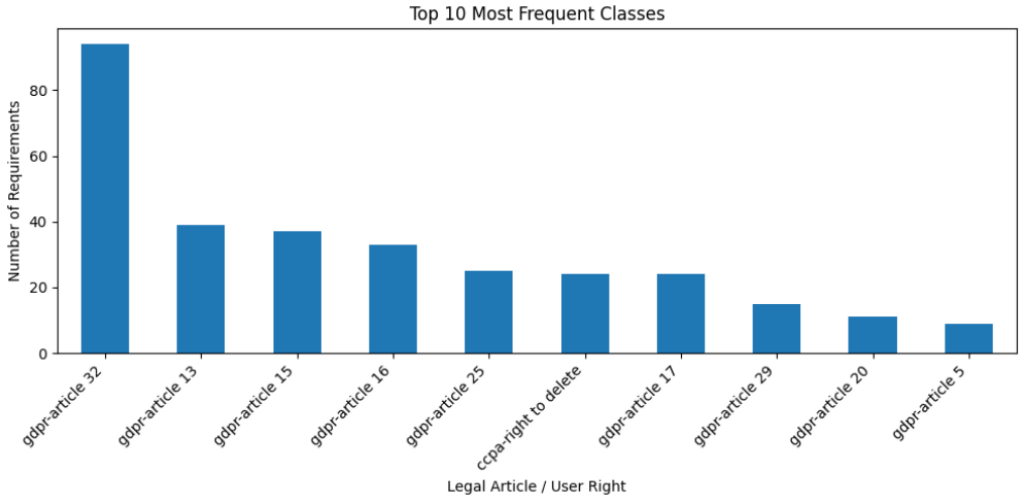

Figure 9: Distribution of the Most Frequent Legal Classes

Although the model demonstrated stable training behaviour throughout the training process, the overall predictive performance remained relatively limited.

Epoch	*Validation Loss*	*Accuracy*	*Macro Precision*	*Macro Recall*	*Macro F1*
1	2.6862	0.0857	0.0511	0.0800	0.0513
2	2.6533	0.1714	0.0959	0.1037	0.0751
3	2.6147	0.2714	0.0957	0.1600	0.1113
4	2.5729	0.3000	0.0997	0.1670	0.1197
5	2.5283	0.3000	0.1012	0.1670	0.1221
6	2.4916	0.2857	0.0968	0.1635	0.1174
7	2.4604	0.3143	0.1702	0.2064	0.1683
8	2.4426	0.3143	0.1702	0.2064	0.1683
9	2.4266	0.3286	0.2399	0.2159	0.1865
10	2.4226	0.3286	0.2399	0.2159	0.1865

Table 9: Validation Performance Across Training Epochs

The validation results show a decrease in validation loss from 2.6862 to 2.4226 at the final epoch, indicating that the model successfully learned semantic patterns from the training data. Similarly, validation accuracy improved from 0.0857 to 0.3286 during training.

Despite this improvement, macro-averaged performance metrics remained low. The final macro-precision, macro-recall and macro-F1 scores reached 0.2399, 0.2159 and 0.1865 respectively. These findings suggest that the model struggled to generalize across all legal categories.

The experiments further revealed that severe class imbalance and semantic overlap significantly affected predictive behaviour. In particular, the model demonstrated strong prediction bias toward highly represented legal categories, especially security-related provisions.

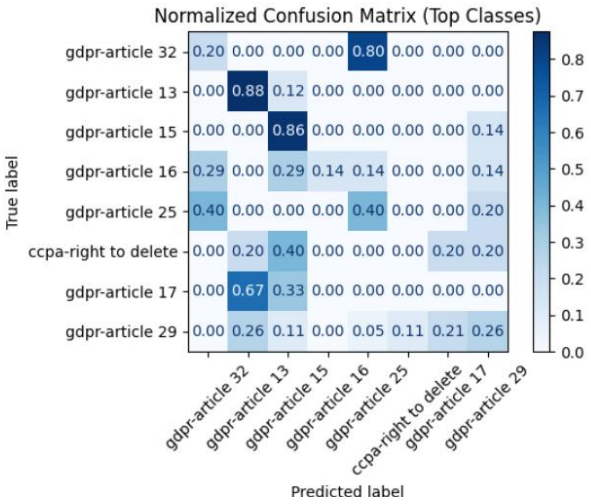

Figure 10: Normalized Confusion Matrix for the Most Frequent Classes

The confusion matrix revealed several misclassification patterns between semantically related legal concepts. In particular, a proportion of Article 13 (Information to be Provided) requirements were incorrectly assigned to Article 32 (Security of Processing), while some Article 15 (Right of Access) requirements were confused with Article 13, reflecting the semantic overlap between access-related and transparency-related requirements. Similarly, a few Article 16 (Right to Rectification) instances were predicted as Article 32. These findings indicate that the model struggled to consistently separate closely related legal concepts, particularly when requirements contained overlapping privacy terminology or implicit legal semantics.

These findings indicate that direct multiclass formulations face difficulty in separating semantically related legal categories within privacy requirements. Although the transformer model was able to capture certain contextual patterns, predictive performance remained limited due to class imbalance, semantic overlap between legal concepts and ambiguity in requirement phrasing.

6.3.2.2 Pairwise Semantic Verification Results

A second experimental approach reformulated the problem as pairwise semantic verification. Instead of directly predicting legal labels, the model evaluated whether a requirement semantically matched a candidate legal concept.

Exp ID	***Negative Sampling (k)***	***Epochs***	***Batch Size***	***Learning Rate***	***Notes***
E1	All negatives	5	16	2e-5	Extreme imbalance

E2	k = 5	5	16	2e-5	Reduced imbalance
E3	k = 3	5	16	2e-5	Better balance
E4	k = 3	3	16	2e-5	Reduced overfitting
E5	k = 3	3	8	2e-5	Better generalization
E6	k = 3	3	8	1e-5	Slower convergence

Table 10: Experimental Configurations for the Pairwise BERT Model

The experiments explored multiple negative sampling strategies, candidate ranking configurations and training settings to analyse their effect on semantic matching performance and prediction stability.

Exp ID	***Accuracy***	***Precision***	***Recall***	***F1-score***	***Key Observation***
E1	> 0.90	0.00–0.05	0.00–0.05	~0.00	Severe bias toward non-match class
E2	0.85–0.88	0.55–0.65	0.50–0.60	0.55–0.62	Improved balance after negative sampling
E3	0.82–0.86	0.70–0.75	0.60–0.65	0.66–0.70	Best trade-off between precision and recall
E4	0.83–0.86	0.72–0.77	0.55–0.62	0.64–0.69	Reduced overfitting with fewer epochs
E5	0.85	0.77	0.62	0.69	Best-performing configuration
E6	0.85	0.77	0.61	0.68	No improvement with lower learning rate

Table 11: Performance of the Pairwise BERT Model Across Experimental Configurations

The experimental results demonstrate that negative sampling played a critical role in improving predictive behaviour. The initial configuration (E1), which used all negative pairs, achieved very high overall accuracy (>0.90) but produced near-zero precision, recall and F1-score values. This behaviour revealed severe prediction bias toward the dominant non-match class caused by extreme class imbalance.

Introducing controlled negative sampling significantly improved performance. Configurations E2 and E3 demonstrated substantial gains in recall and F1-score, indicating improved balance between positive and negative semantic matches. Among these configurations, E3 achieved the strongest trade-off between precision and recall, reaching an F1-score between 0.66 and 0.70.

Additional experiments further explored the effect of training duration, batch size and learning rate. Reducing the number of training epochs in E4 helped reduce overfitting while maintaining comparable overall performance. The best-performing configuration was obtained in E5, which achieved an F1-score of 0.69 together with precision and recall values of 0.77 and 0.62 respectively. In contrast, lowering the learning rate in E6 did not produce further improvements and resulted in slower convergence behaviour.

Overall, the results demonstrate that pairwise semantic verification improved contextual semantic matching and reduced certain forms of label confusion compared to the multiclass formulation. This approach reduced direct competition between similar legal labels by evaluating each requirement-label pair separately.

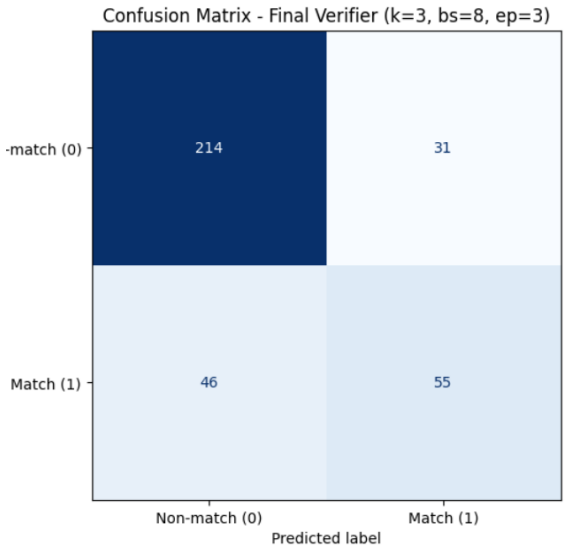

Figure 11: Confusion Matrix for the Pairwise BERT Verifier

The confusion matrix further illustrates the behaviour of the final verifier. The model correctly identified many non-match pairs, producing 214 true negatives while generating relatively few false positives (31). This indicates that the verifier was generally effective at rejecting semantically unrelated requirement-label pairs.

At the same time, the model produced 46 false negatives compared to 55 true positive matches, suggesting that several legal relationships were still missed during verification. This behaviour was particularly evident in cases where multiple legal rights were simultaneously applicable or where the distinction between related legal concepts remained ambiguous.

Overall, these findings indicate that although pairwise semantic verification improved contextual matching quality and reduced certain forms of label confusion, final predictions remained sensitive to candidate ranking quality and to semantic similarity between related legal concepts.

6.3.2.3 Prompt-based LLM Results

Prompt-based experiments were conducted to evaluate the ability of LLMs to perform zero-shot mapping between privacy requirements and GDPR or CCPA legal concepts. The experiments specifically focused on analysing the effect of prompt structure, semantic guidance and output constraints on prediction quality and output stability.

The evaluation was performed using Meta Llama-3.2-3B [18] on a manually validated subset of 100 privacy-related requirements. Each requirement was associated with a label corresponding to either a GDPR article or a CCPA right.

The first experimental configuration employed a minimal zero-shot formulation without additional semantic guidance or output constraints.

Prompt 1:

“Classify the following privacy requirement to an appropriate GDPR article or CCPA right.
Return only one label (for example: "gdpr-article 17" or "ccpa-right to delete").
Requirement: {req}”

This prompt produced highly unstable outputs and extremely poor predictive performance. In many cases, the model generated explanatory text instead of valid labels, returned labels outside the predefined taxonomy or failed to follow the requested output format. These findings indicate that minimal zero-shot instructions were insufficient for performing legal classification.

The second prompt introduced role-based reasoning guidance and instructed the model to first identify the underlying privacy-related concept before selecting the corresponding legal provision

Prompt 2:

“You are a data privacy expert familiar with GDPR and CCPA.
Step 1: Identify the type of user right expressed in the requirement for example
access, deletion, correction, portability, objection, consent, security, transparency,

breach notification.
Step 2: Map it to an appropriate GDPR article or CCPA right.
Return the label in the following format:
• gdpr-article X
• ccpa-right ...
Requirement: {req}”

Although the reasoning-oriented formulation slightly improved semantic understanding, output consistency remained problematic. The model frequently produced verbose responses, incomplete outputs or formatting deviations, reducing the overall reliability of the generated predictions.

The third prompt introduced explicit output constraints by forcing the model to select a label from a predefined legal taxonomy.

Prompt 3:
“You must select exactly correct label.
Possible outputs: [predefined list of GDPR and CCPA labels used in the experiment]
Rules:
• Return a label from the list.
• Do not explain.
• Do not output anything else.
Requirement: {req}”

The constrained-output formulation significantly improved output stability and reduced invalid predictions. By restricting the response space to predefined categories, the model generated more consistent predictions and fewer formatting violations.

The fourth prompt extended the constrained formulation by embedding semantic descriptions and explicit legal decision rules directly within the prompt.

Prompt 4:
“You are a GDPR and CCPA expert.
Your task is to map the following software privacy requirement to exactly one legal label.
[semantic descriptions of GDPR and CCPA labels used in the experiment]
Rules:

1. Select exactly one label.
2. Prefer the most specific user right.
3. Use gdpr-article 17 for deletion or erasure requests.
4. Use gdpr-article 15 or ccpa-right to access for data access requests.
5. Use gdpr-article 16 for correction or rectification requests.
6. Use gdpr-article 20 for data portability requests.
7. Use gdpr-article 21 for objection or opt-out of processing.
8. Use gdpr-article 22 for automated decision-making or profiling.
9. Use gdpr-article 13/14 or ccpa-notice at collection for transparency or information provision.
10. Use gdpr-article 32 for security-related requirements.
11. Use gdpr-article 25 for privacy by design or default settings.
12. Use gdpr-article 33/34 for data breach notification.
Requirement: {req}”

This configuration achieved the strongest overall performance among all evaluated prompts. The inclusion of semantic descriptions and explicit decision rules improved contextual understanding and reduced certain forms of semantic ambiguity, particularly for deletion, access and portability-related requirements.

Finally, a fifth prompt explored a more complex formulation based on staged reasoning and structured JSON output generation.

Prompt 5:
“You are a GDPR and CCPA expert.
Stage 1: classify into category
Stage 2: select final label
Return JSON:
{"category": "...",
"predicted_label": "...",
"confidence": 0.0,
"rationale": "..."}
Requirement: {req}”

Although this formulation improved output structure by requiring intermediate reasoning and structured responses, it introduced substantial output instability. The

model frequently produced malformed JSON structures, incomplete outputs and invalid label formats. These findings suggest that increased prompt complexity introduced additional instruction-following difficulties, as the model was required to perform semantic reasoning, confidence estimation and strict formatting adherence.

The overall performance of the prompt-based experiments is summarized in Table 11.

Prompt	*Accuracy*	*Macro Precision*	*Macro Recall*	*Macro F1*	*Weighted F1*
Prompt 4	0.10	0.042	0.039	0.040	0.061
Prompt 3	0.06	0.028	0.023	0.025	0.048
Prompt 5	0.04	0.032	0.027	0.029	0.052
Prompt 2	0.02	0.027	0.025	0.026	0.049
Prompt 1	0.01	0.018	0.017	0.017	0.041

Table 12: Performance of prompt-based LLM approaches across evaluation metrics.

The results presented in Table 12 demonstrate that different prompt formulations significantly affected the ability of the model to map requirements to legal labels. Among all evaluated configurations, Prompt 4 achieved the highest performance, reaching an accuracy of 0.10 and the highest macro F1-score (0.040). The performance of Prompt 4 suggests that semantic descriptions and explicit legal decision rules within the prompt helped the model better distinguish between overlapping legal concepts and reduced semantic ambiguity during label selection.

Prompt 3 achieved the second-best overall accuracy (0.06), despite using a simpler formulation compared to Prompt 4. This finding indicates that strict output constraints alone improved prediction reliability by reducing formatting variability and limiting the response space to predefined legal labels.

Although Prompt 5 introduced staged reasoning and structured JSON outputs, its performance remained relatively limited, achieving only 0.04 accuracy. This behaviour suggests that increasing prompt complexity does not necessarily improve classification performance and may instead introduce additional challenges.

Prompts 1 and 2 demonstrated the weakest performance across all evaluation metrics. Prompt 1 achieved only 0.01 accuracy, confirming that minimal zero-shot prompting is insufficient for legal mapping tasks involving highly overlapping privacy concepts.

Overall, the results indicate that semantic guidance and output control substantially improve zero-shot legal classification behaviour. However, even the best-performing prompt remained significantly below the performance achieved by the supervised and hybrid approaches evaluated in other sections.

An additional evaluation focused on output validity and prediction stability.

Prompt	*Correct Predictions*	*Total Samples*	*Invalid Predictions*
Prompt 4	10	100	37
Prompt 3	6	100	5
Prompt 5	4	100	85
Prompt 2	2	100	90
Prompt 1	1	100	95

Table 13: Prediction correctness and output stability across prompt configurations.

Table 13 further highlights the relationship between prompt complexity and output stability. Prompt 3 achieved the lowest number of invalid predictions (5), demonstrating that strict output constraints were highly effective in enforcing valid label generation and reducing formatting violations.

In contrast, Prompt 1 and Prompt 2 produced extremely high numbers of invalid outputs, indicating that loosely constrained prompts frequently caused the model to generate explanatory text, hallucinated labels or incomplete responses instead of valid predictions.

Prompt 5 exhibited 85 invalid predictions despite incorporating structured reasoning and JSON formatting. This suggests that the combination of reasoning steps and strict formatting constraints negatively affected output reliability.

Overall, the results indicate that prompts with stronger semantic guidance improved contextual understanding, but more complex formatting and reasoning instructions often reduced output stability and increased invalid predictions.

The confusion matrix of the best-performing configuration (Prompt 4) further revealed substantial confusion between semantically related legal concepts.

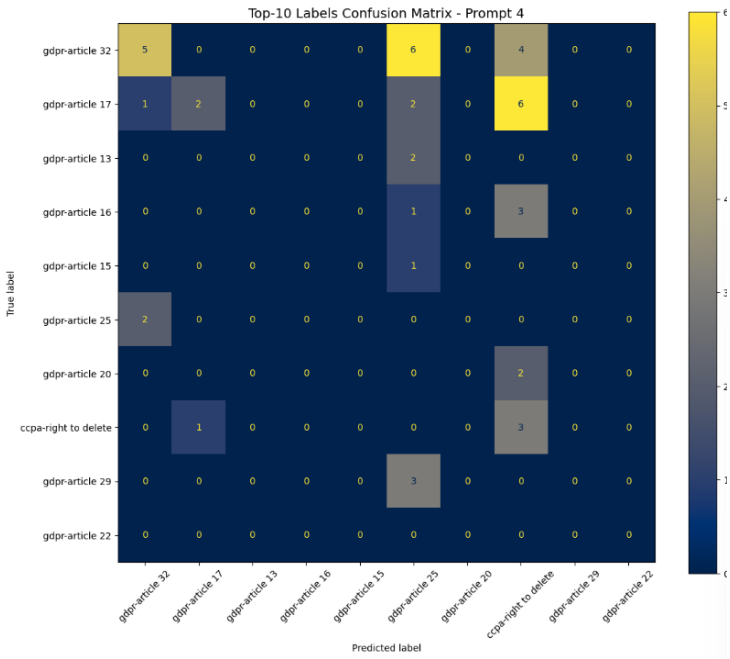

Figure 12: Top 10 labels confusion matrix for Prompt 4.

Figure 12 presents the confusion matrix for the top 10 labels obtained from the Prompt 4. The matrix reveals that although certain labels are correctly predicted, substantial confusion remains between semantically related legal categories.

A particularly strong prediction bias can be observed toward GDPR Article 32, which frequently appears both as a correct prediction and as a dominant misclassification target. Several requirements were incorrectly mapped to Article 32, indicating that the model often overgeneralized security-related terminology.

Confusion is also evident between GDPR Article 17 and the CCPA Right to Delete, as well as between GDPR Articles 32 and 25. These patterns suggest that the model struggled to distinguish between legal concepts that share overlapping vocabulary.

The confusion matrix additionally demonstrates that several labels were rarely predicted, indicating that the model tended to favour a limited subset of dominant legal categories during uncertain predictions. Similar sensitivity to prompt structure and prediction variability has also been discussed in recent prompt engineering literature [19].

Overall, the confusion matrix highlights that although Prompt 4 improved overall semantic grounding, the model continued to experience substantial difficulty

performing fine-grained differentiation between related GDPR and CCPA legal provisions.

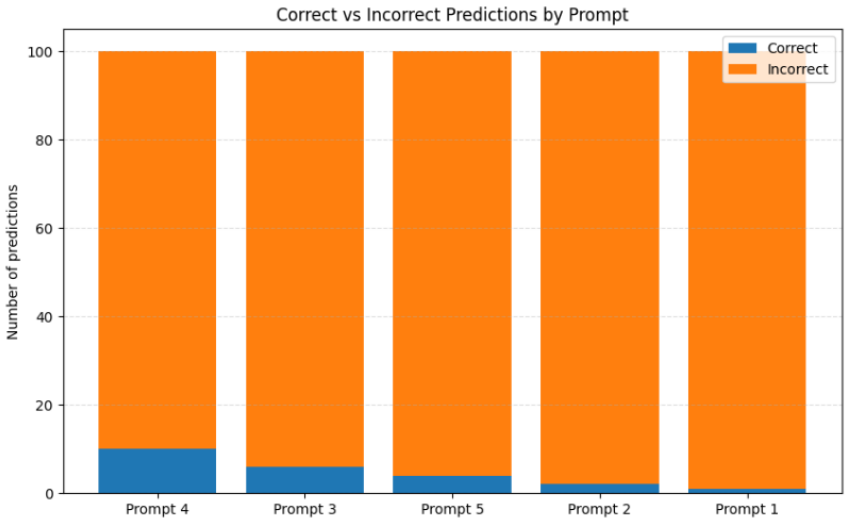

Figure 13: Distribution of correct and incorrect predictions across different prompt configurations.

Figure 13 summarizes the overall prediction behaviour across all prompt configurations by comparing the number of correct and incorrect predictions. The figure clearly illustrates that incorrect predictions dominate across all prompts, emphasizing the overall difficulty.

Overall, the experiments showed that prompt structure strongly affected both prediction quality and output stability. Prompts containing semantic guidance, legal decision rules and structured output constraints produced more reliable predictions than simpler zero-shot prompts. However, overall performance remained limited, suggesting that zero-shot prompting alone is insufficient for accurate legal rights mapping.

6.3.3 Results of the Proposed Hybrid Pipeline

Based on the limitations observed in the previous experimental approaches, the final Stage 2 system adopted a hybrid multi-label architecture integrating lexical, semantic and rule-based components for legal rights mapping. The objective of this design was to improve prediction stability and semantic consistency under conditions of class imbalance, overlapping legal concepts and limited annotated training data.

Dataset	*Micro Precision*	*Micro Recall*	*Micro F1*	*Macro Precision*	*Macro Recall*	*Macro F1*
VAL	0.699	0.689	0.694	0.369	0.350	0.349
TEST	0.706	0.658	0.681	0.326	0.275	0.283

Table 14: Performance of the Final Hybrid Architecture Across Validation and Test Sets

Table 14 presents the overall performance of the final architecture on both the validation and test sets using micro- and macro- evaluation metrics. The results demonstrate that the proposed architecture achieved the most balanced and stable overall performance among all evaluated approaches. The relatively small performance difference between the validation and test sets indicates that the model generalizes reasonably well and does not exhibit severe overfitting behaviour.

The model achieved a micro-F1 score of 0.694 on the validation set and 0.681 on the test set, demonstrating relatively stable predictive behaviour across unseen data. Micro-averaged precision remained slightly higher than recall, indicating that the model generates fewer incorrect label assignments than missed relevant labels. This behaviour suggests that the final prediction mechanism is more likely to exclude uncertain labels than include potentially unrelated ones, although some valid legal concepts may therefore remain unselected when multiple mappings are possible.

In contrast, macro-averaged performance remained lower, with macro-F1 values of 0.349 on the validation set and 0.283 on the test set. This discrepancy reflects the imbalanced nature of the dataset, where a limited number of dominant legal categories appear more frequently than others. As a result, the model performs better on frequent labels while struggling with underpresented categories. This behaviour is common in multi-label classification problems with imbalanced datasets, where learning tends to become biased toward dominant classes [20][21].

The strong micro-level performance suggests that the combination of lexical TF-IDF features and Sentence-BERT embeddings successfully captures both explicit legal terminology and deeper contextual relationships between requirements and legal concepts. In particular, the semantic representation layer improved the ability of the model to generalize beyond exact keyword matching, allowing semantically similar requirements to be mapped even when they used different linguistic expressions. At the

same time, the rule-based mechanisms helped correct certain prediction errors and improve the handling of overlapping legal concepts.

Dataset	*K*	*Recall*	*Coverage*
VAL	1	0.7015	0.6716
VAL	2	0.8209	0.8035
VAL	3	0.9403	0.9279
VAL	5	0.9552	0.9478
TEST	1	0.7200	0.6633
TEST	2	0.8100	0.7700
TEST	3	0.8800	0.8517
TEST	5	0.9000	0.8917

Table 15: Ranking Performance Across Different Numbers of Candidate Labels

To further analyse the semantic ranking capability of the model independently of the final label selection mechanism, recall and coverage metrics were computed for multiple values of k. As shown in Table 15, the ranking-based evaluation demonstrated that the correct legal labels frequently appeared among the highest-ranked candidate predictions across both datasets.

When only the highest-ranked candidate label was considered, recall reached 0.7015 on the validation set and 0.7200 on the test set. This indicates that in most cases the top-ranked prediction corresponded to at least one correct legal concept. Performance reached 0.9403 on the validation set and 0.8800 on the test set when the three highest-ranked predictions were included. Similarly, coverage steadily increased as more candidate labels were considered, indicating that the model was able to recover a significant proportion of the complete ground-truth label set within its top-ranked predictions.

These findings provide important insight into the behaviour of the hybrid architecture. In many cases, the correct legal labels were already present among the top-ranked candidates but were not included in the final predicted set. This suggests that the semantic representation layer itself performs effectively and successfully captures meaningful contextual relationships between requirements and legal concepts.

Consequently, the main limitation of the system appears to lie in the final label selection process rather than in the identification of relevant legal concepts.

The analysis additionally demonstrates that the model successfully captures meaningful relationships between software requirements and legal concepts despite the complexity of the task. This behaviour is important in legal requirement mapping scenarios, where a single requirement may correspond to multiple related legal concepts. In such cases, the ability of the model to rank semantically relevant labels highly provides additional evidence that the underlying legal context of the requirement has been successfully identified.

True Label	***Predicted Instead***	***Count***
gdpr-article 15	gdpr-article 32	3
gdpr-article 32	gdpr-article 29	2
gdpr-article 17	gdpr-article 5	1
ccpa-right to delete	gdpr-article 5	1
gdpr-article 25	gdpr-article 32	1

Table 16: Most frequent label confusions observed in the test set

To further investigate the types of prediction errors produced by the system, the most common semantic confusion patterns between legal concepts were analysed. Table 16 presents the most frequent label confusions observed on the test set.

The confusion analysis reveals that most prediction errors occur between semantically related legal concepts. The most frequent confusion was observed between Right of Access (GDPR Article 15) and Security of Processing (GDPR Article 32). This behaviour suggests that requirements involving access to personal data are often semantically associated with security-related functionality, particularly when expressions related to authentication, authorization or controlled access appear within the requirement description. As a result, the model prioritizes security-related articles over user access rights.

A similar semantic overlap was observed between Security of Processing (GDPR Article 32) and Processing Under the Authority of the Controller or Processor (GDPR Article 29), both of which involve restrictions regarding who is authorized to process or access personal information. Since these concepts frequently share common

terminology related to restricted access and authorized processing, the model struggles to fully distinguish their legal intent based solely on textual semantics.

Deletion-related legal concepts also produced notable confusion patterns. Specifically, the Right to Erasure (GDPR Article 17) and the CCPA Right to Delete were occasionally misclassified as Principles Relating to Processing of Personal Data (GDPR Article 5). This behaviour suggests that the model successfully captures the broader semantic context of data lifecycle management, including retention, deletion and storage operations, but struggles to distinguish between high-level data processing principles and explicit user rights. Similarly, Privacy by Design and by Default (GDPR Article 25) was sometimes confused with security-related provisions because such requirements are often expressed indirectly without explicit privacy-related terminology.

The confusion patterns indicate that most prediction errors occurred between semantically related privacy concepts. The analysis also highlights the difficulty of distinguishing between legal rights that use similar terminology or describe related privacy actions.

Despite the remaining challenges associated with semantic ambiguity, class imbalance and multi-label prediction completeness, the hybrid pipeline outperformed the previous experimental approaches by producing more stable predictions across different legal categories. The integration of semantic embeddings, lexical representations, prototype similarity features and rule-based mechanisms provided a more effective framework for legal rights mapping than purely transformer-based or prompt-based alternatives.

The evaluation demonstrates that the proposed hybrid architecture is capable of effectively aligning natural language software requirements with legal privacy concepts. At the same time, the results highlight several important directions for future improvement, including better handling of underrepresented labels, refinement of the final label selection process and improved differentiation between closely related legal concepts.

Chapter 7 – Web Application

7.1 Overview of the Developed Tool

The developed application is a web-based system designed to assist in the identification and analysis of privacy requirements in software documentation. It provides an interactive environment through which users can submit requirements, review results and obtain structured information related to privacy concepts and user rights.

The analysis process is organized into two stages. In the first stage, each requirement is examined to determine whether it is privacy related. The results of this stage are presented to the user in a dedicated review interface, where each requirement is clearly labelled and can be inspected individually.

A key feature of the application is the inclusion of a user validation step between the two stages. Instead of automatically proceeding to further analysis, users are given the opportunity to confirm or modify the system's initial classification decisions. This design choice acknowledges that privacy-related meaning in natural language can be ambiguous and context-dependent and allows users to retain control over the analysis process.

User modifications to the initial classification results are stored as feedback data for future dataset enrichment and model retraining. This mechanism enables the gradual collection of corrected examples that may support future improvements in prediction quality and system adaptation.

Once the user confirms the selected requirements, the system proceeds to the second stage, where each requirement is associated with relevant privacy concepts. These concepts represent privacy obligations and user rights and are presented in a structured format. When multiple concepts correspond to similar privacy meanings, they are grouped together to avoid redundancy in the presentation.

The results are displayed in a comprehensive and organized view, including the original requirement text and the identified privacy concepts. All results can be exported in CSV format for further analysis or documentation purposes.

The application is intended for use by developers, requirements engineers, privacy analysts, and researchers who seek structured support in examining software requirements from a privacy perspective.

Overall, the application aims to provide a practical and user-friendly environment for supporting the analysis of privacy requirements and facilitating privacy compliance activities during software development processes.

7.2 System Functionality

The application provides a user-friendly web interface that enables users to submit and analyse software requirements with respect to privacy considerations. Users can enter requirements directly into a text area, with each requirement written on a separate line, allowing the system to process both individual inputs and multiple requirements in batch form.

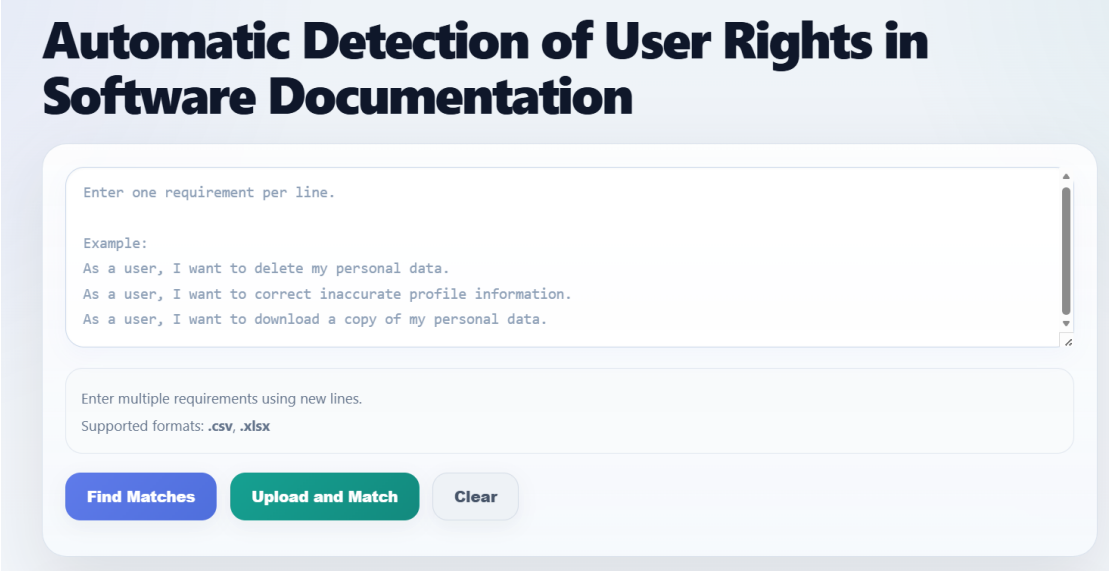

Figure 14: Input interface of the application.

To assist users in formatting their input correctly, the interface includes example requirements within the text area. These examples illustrate the expected structure of input.

In addition to manual input, the system supports file-based input through an upload functionality. Users can upload requirement datasets in formats such as .csv and .xlsx, enabling efficient processing of larger collections of requirements. This feature is

particularly useful in practical scenarios where requirements are stored in structured documents.

The interface provides three main actions. The “Find Matches” option triggers the analysis of manually entered requirements, the “Upload and Match” option processes requirements from an uploaded file and the “Clear” option resets the interface and removes any previously entered data.

Once the input is submitted, the system proceeds to the first stage, where each requirement is evaluated for its relevance to privacy. The results are presented in an interactive review interface, where each requirement is displayed alongside its classification. Requirements identified as privacy-related are automatically selected, indicating that they will be considered for further analysis.

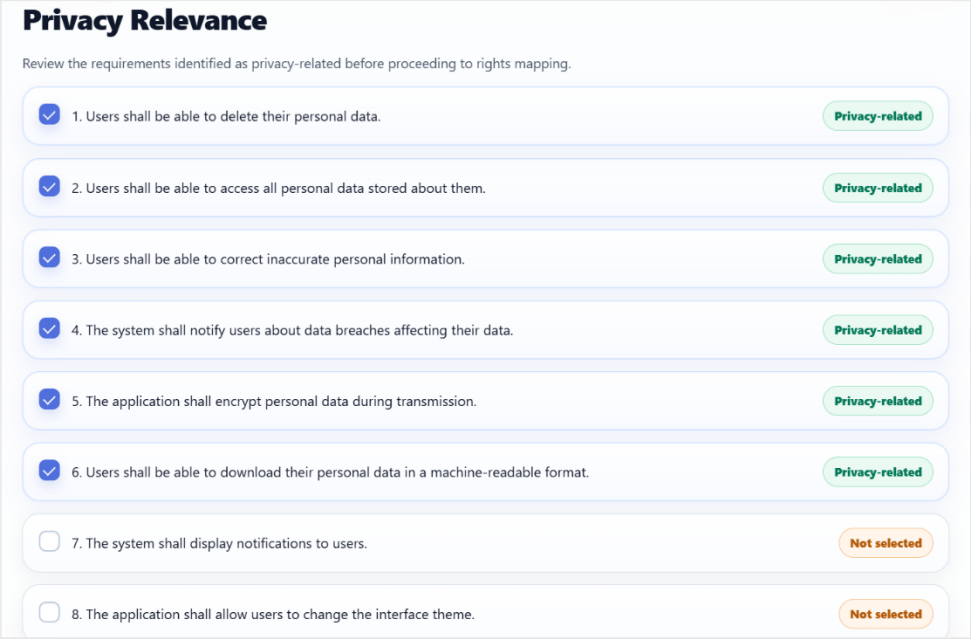

Figure 15: Privacy relevance review interface

As illustrated in Figure 15, requirements that involve personal data operations are typically identified as privacy related. This behaviour demonstrates the system’s ability to distinguish between privacy-relevant and non-relevant requirements.

A key feature of this stage is the ability for users to manually adjust the system’s selections. Through the interface, users can select or deselect requirements, effectively validating or correcting the system’s initial decisions. This interaction allows users to handle ambiguous cases and ensures that only relevant requirements proceed to the next stage.

After user validation, the system generates structured results for each selected requirement. These results are presented through an organized interface that displays the identified privacy concepts in a clear manner. Each concept is accompanied by a concise title and a short description, helping users understand its meaning without requiring prior legal expertise.

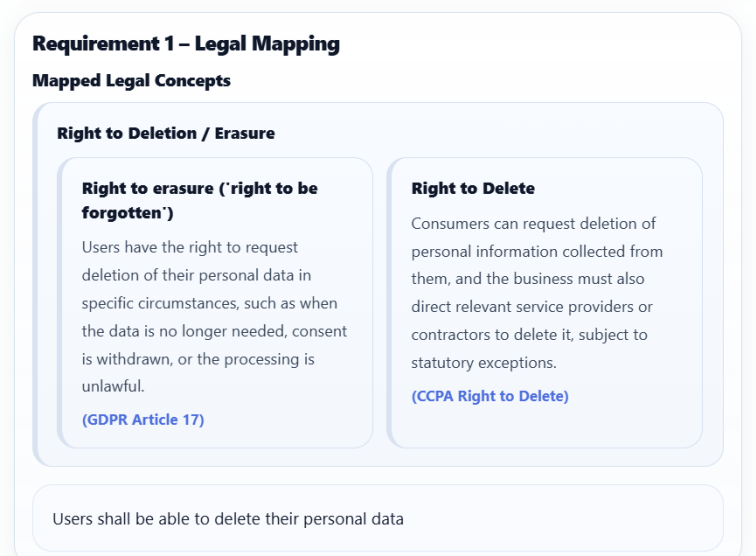

Figure 16: Example of legal mapping results.

When multiple concepts are closely related, they are grouped together within a unified card. This design avoids redundancy and improves readability by presenting related concepts in a single, consolidated view. For example, equivalent rights identified across different regulatory frameworks are displayed together, allowing users to easily recognize their relationship.

Finally, the system supports exporting the results in CSV format, enabling users to store, review and further process the identified privacy-related requirements and associated legal concepts outside the platform.

The application provides a structured and interactive process for analysing privacy requirements and presenting the generated results in a clear and organized manner.

Chapter 8 – User Feedback on the Web Application

8.1 Questionnaire Design

To gather user feedback on the developed application, an online questionnaire was distributed to participants. The objective of this feedback study was to examine how users perceive the system in terms of ease of use, correctness of results, understanding of privacy-related requirements and overall trust in the platform. In addition, the evaluation aimed to collect qualitative feedback regarding potential limitations and areas for improvement.

The questionnaire was structured into three main parts. The first part collected basic background information about the participants, including their occupation and their level of experience with data privacy laws and software requirements. This information was used to better understand the profile of the participants and analyse their responses accordingly.

The second part focused on collecting participant feedback regarding the application. Participants were asked to rate a series of statements using a 5-point scale, ranging from “Strongly disagree” (1) to “Strongly agree” (5). The statements assessed key aspects of the system, including its ease of use, its usefulness in helping users understand whether a requirement is related to user rights, the perceived correctness of the identified user rights, the level of trust in the system’s results and the overall user satisfaction. In addition to these ratings, participants were also asked to provide brief explanations in cases where they observed incorrect results or when elaborating on whether they would trust the platform.

The third part included open-ended questions, allowing participants to suggest possible improvements to the application. This qualitative feedback provided additional insights into usability issues, limitations of the system’s predictions and opportunities for further enhancement.

The questionnaire was conducted anonymously, and no personal data were collected. Participation was voluntary and respondents provided informed consent before completing the questionnaire. The collected data were used exclusively for academic research purposes, in line with the objectives of this thesis.

8.2 Participants Profile

A total of 16 participants completed the questionnaire. The participants represented different backgrounds and levels of familiarity with privacy legislation and software requirements engineering.

As illustrated in Figure 17, most participants were students (62.5%), while researchers and software developers each represented 18.8% of the sample.

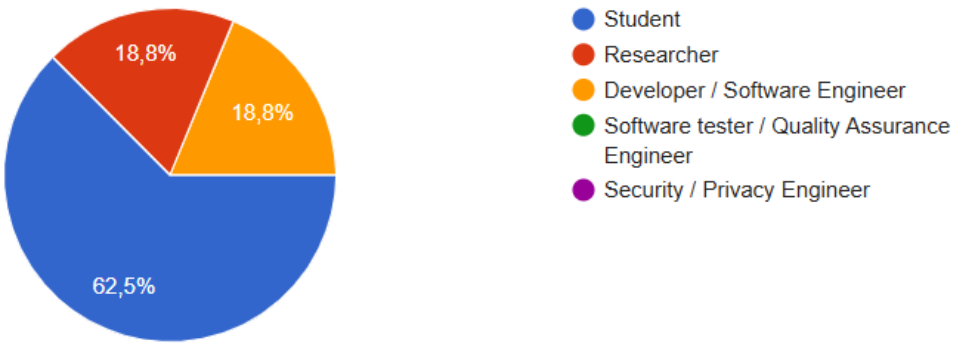

Figure 17: Distribution of participants by occupation.

The questionnaire additionally examined participants' familiarity with privacy regulations such as GDPR and CCPA. As shown in Figure 18, most participants reported relatively limited prior experience with privacy legislation, with the majority selecting lower familiarity levels.

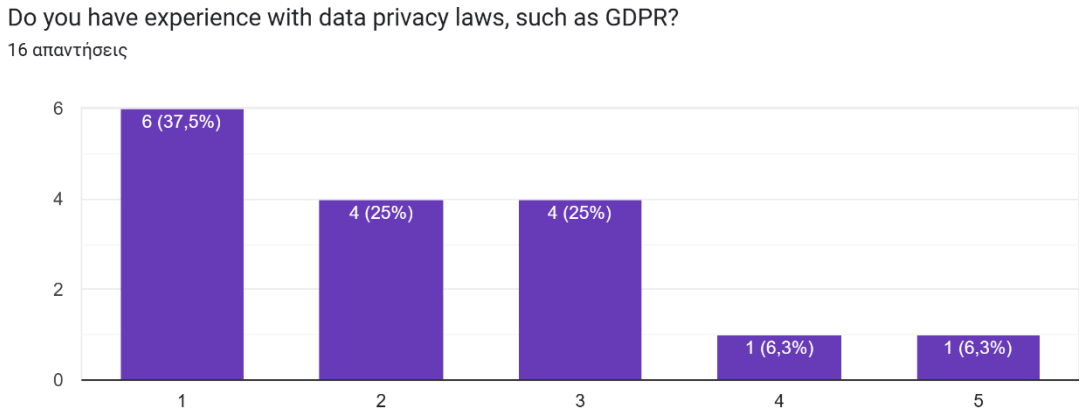

Figure 18: Participant familiarity with privacy regulations.

Participants were also asked about their experience working with software requirements and user stories. The responses indicate that several participants had moderate to high familiarity with software requirements engineering concepts.

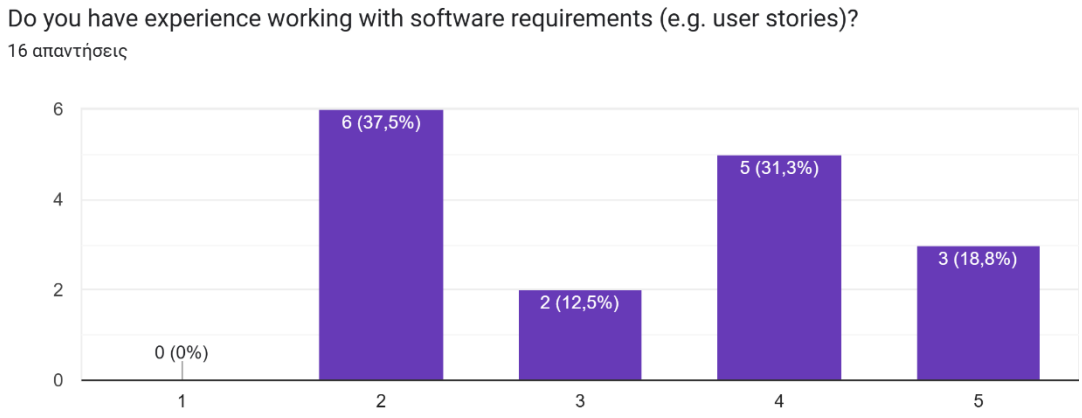

Figure 19: Participant familiarity with software requirements and user stories.

8.3 Questionnaire Results

The questionnaire results indicate a highly positive overall perception of the developed application across multiple usability and reliability dimensions.

Participants reported very high levels of usability. As illustrated in Figure 20, 87.5% of participants strongly agreed that the application was easy to use, while the remaining 12.5% agreed with the statement.

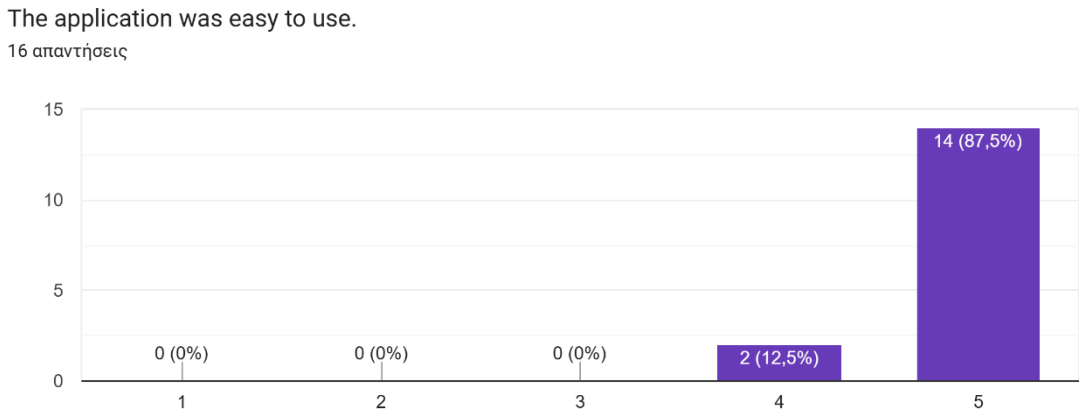

Figure 20: Results for application usability.

The collected feedback further showed that the system effectively helped users understand whether software requirements were related to privacy rights and legal concepts. As shown in Figure 21, most participants either agreed (37.5%) or strongly agreed (56.3%) that the application improved their understanding of privacy-related requirements, while only one participant selected a neutral response.

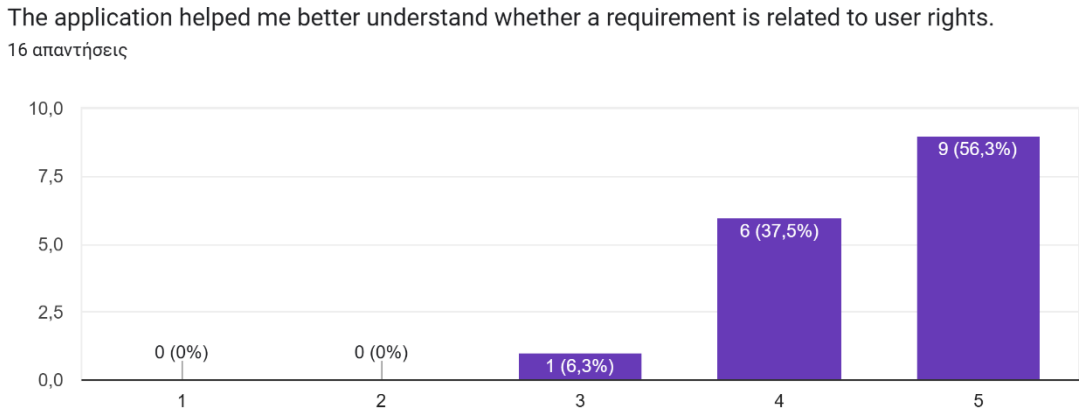

Figure 21: Results for requirement understanding support.

Participants also expressed positive opinions regarding the correctness of the identified legal mappings. As illustrated in Figure 22, 62.5% of respondents strongly agreed that the results were mostly correct, while the remaining 37.5% agreed. No negative evaluations were reported.

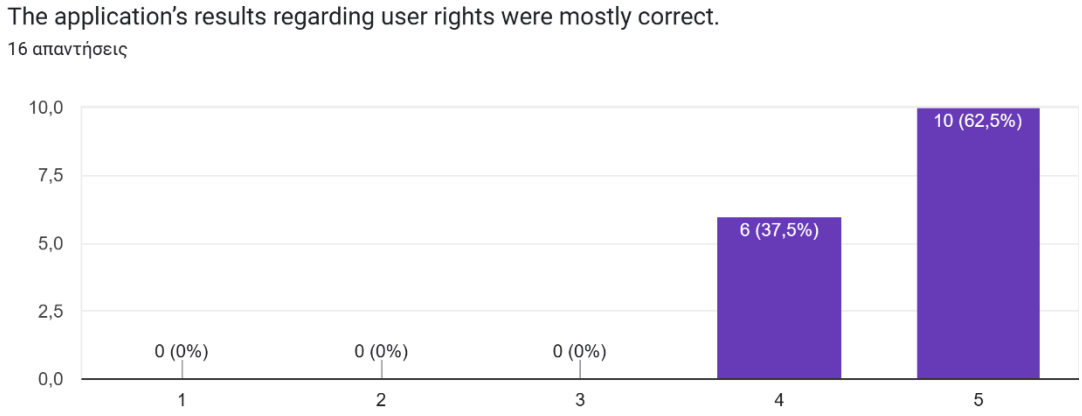

Figure 22: Responses regarding the correctness of the identified legal mappings.

Similarly, users reported high levels of trust in the platform. Figure 23 shows that most participants either agreed (31.3%) or strongly agreed (62.5%) that they trusted the

results produced by the application. Only one participant provided a neutral response, while no participants expressed disagreement.

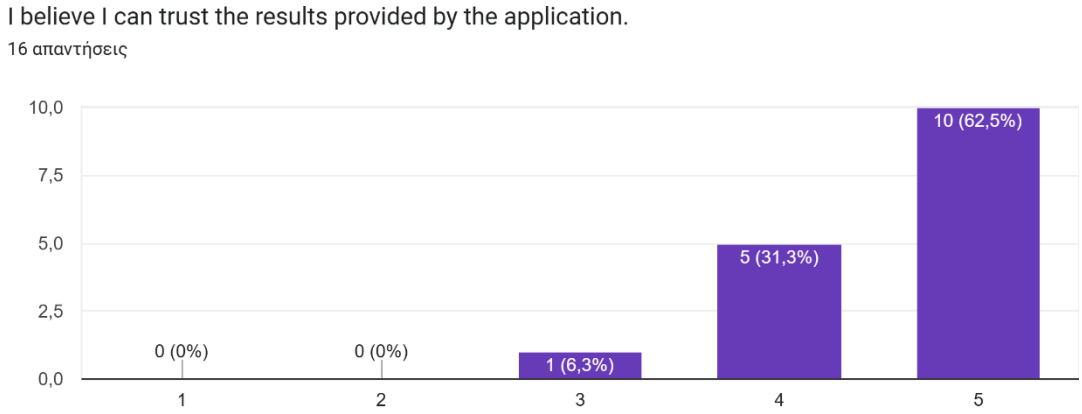

Figure 23: Participant responses regarding trust in the generated results.

The results additionally indicated high overall user satisfaction. As presented in Figure 24, 81.3% of participants strongly agreed that they were satisfied with the application, while the remaining 18.8% agreed with the statement.

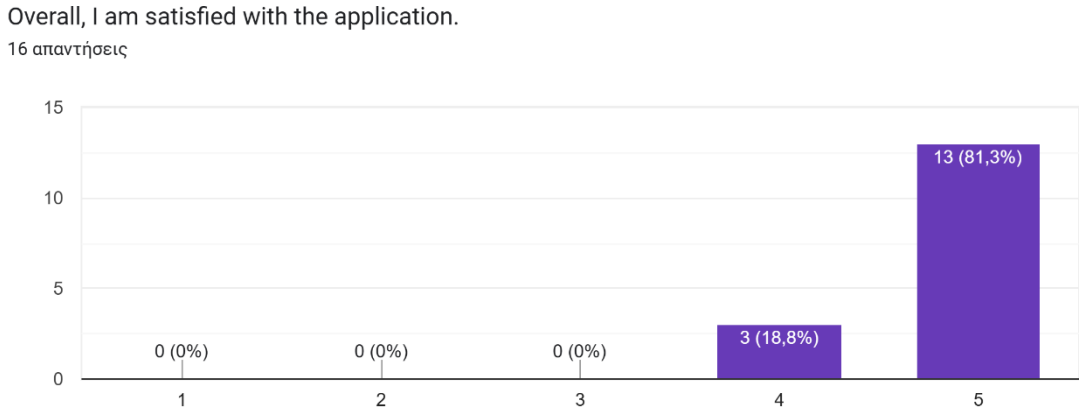

Figure 24: Overall user satisfaction with the application.

The quantitative results suggest that users perceived the application as usable and reliable. The consistently high agreement levels across all evaluation dimensions indicate that the proposed system provides useful support for analysing software requirements in relation to privacy legislation.

8.4 Qualitative Feedback and Discussion

Participants were additionally asked to provide qualitative feedback regarding incorrect predictions, trust in the platform and possible improvements. Overall, the responses indicated that most observed limitations were related to semantic ambiguity, overlapping legal concepts and the distinction between privacy and security requirements.

Several participants reported cases where requirements were associated with legal rights that were not directly relevant to the intended privacy concept. For example, one participant noted that the requirement *"the system should collect personal data for specified purposes"* was incorrectly mapped to GDPR Article 21 (Right to Object) and the CCPA Right to Opt-Out, although the requirement was more closely related to purpose limitation and data processing principles. Similarly, the same participant observed that the requirement *"the system should implement appropriate technical and organizational security measures"* was incorrectly associated with GDPR Article 22 concerning automated decision-making, despite the requirement being primarily security oriented.

Multiple participants also identified confusion between privacy-related access rights and security-related access control mechanisms. Requirements involving restricted or role-based access to personal data were sometimes associated with GDPR Article 15 (Right of Access), even though these cases mainly referred to security of processing under GDPR Article 32. Participants additionally noted cases involving incomplete or overly broad legal mappings, where the system either failed to assign all relevant concepts or generated additional partially related labels.

Some comments further highlighted the difficulty of identifying implicit or context-dependent privacy meaning. For example, one participant observed that the requirement *"the app should personalize content based on user behaviour"* was associated with GDPR Article 22 concerning automated decision-making, even though personalization does not always necessarily imply profiling or automated decision-making under GDPR provisions.

Despite these limitations, most participants stated that most of the requirements were correctly classified and mapped. Several respondents emphasized that the platform was particularly useful as a supportive analysis tool for understanding the relationship

between software requirements and legal privacy concepts, especially for non-expert users. At the same time, some participants noted that the system should primarily be used as a decision-support tool rather than as a fully autonomous legal analysis system, particularly in ambiguous or legally complex cases.

Participants also provided several suggestions for future improvement. The most common recommendation involved reducing overly broad legal mappings and improving the distinction between closely related legal concepts. Additional suggestions included expanding the training dataset, improving semantic reasoning capabilities and providing clearer explanations describing why specific legal concepts were selected.

Some respondents additionally proposed technical improvements related to system implementation and performance, including frontend optimizations, improved API response handling and stronger server-side error management. One participant also noted that certain Stage 2 analysis operations occasionally required noticeable processing time.

The qualitative feedback suggests that users perceived the platform as useful, understandable and practically valuable, while also highlighting remaining challenges. These observations are consistent with the experimental findings presented in Chapter 6 and further support the importance of combining automated analysis with explainability and human validation mechanisms.

Chapter 9 - Conclusion and Future Work

9.1 Conclusion

This thesis presented a privacy analysis pipeline for the automatic detection and legal mapping of privacy requirements. The proposed approach combines machine learning, semantic analysis and rule-based techniques to support the detection of privacy requirements and their alignment with relevant user rights.

The proposed system consisted of two main stages, the detection of privacy requirements and their mapping to relevant user rights. Overall, different techniques were explored and combined across both stages of the system to improve the effectiveness and reliability of the results.

In addition, a web-based application was developed to provide an environment for analysing software requirements and presenting the generated results in an interactive and user-friendly manner.

The evaluation results demonstrated that the system was capable of effectively analysing realistic software requirements. At the same time, the evaluation process highlighted several challenges associated with privacy requirement analysis. The feedback collected during the user evaluation further indicated that the platform can provide useful support for software development and privacy compliance activities.

The thesis demonstrates the potential of automated approaches for analysing privacy requirements and supporting regulatory compliance in software engineering environments.

9.2 Future Work

Although the proposed system achieved encouraging results, several opportunities for future improvement remain.

One possible direction concerns the expansion of the training dataset. While the current system was evaluated using annotated privacy requirements and user-based evaluation procedures, future work could include larger-scale experiments using additional software requirements datasets. Such evaluation could provide deeper insights into the generalization capability of the proposed pipeline.

Another possible extension involves expanding the supported legal frameworks beyond GDPR and CCPA. Additional privacy regulations such as the CPRA or other international privacy laws could be incorporated to support broader regulatory compliance analysis.

The current system focuses on textual requirements expressed in natural language. Future research could investigate the analysis of additional software artefacts, including use case specifications, design documents, issue reports and technical documentation generated during software development processes.

The proposed approach provides a foundation for future improvements and extensions related to automated privacy requirement analysis and regulatory compliance support in software engineering environments.

Bibliography

[1] S. L. Spósito, J. F. G. Targino, G. R. S. Silva, L. Peotta, D. P. Porto, F. L. L. Mendonça, and E. D. Canedo, “A Comprehensive Review of Techniques, Methods, Processes, Frameworks, and Tools for Privacy Requirements,” Journal of Internet Services and Applications, vol. 16, no. 1, 2025.

[2] F. Casillo, V. Deufemia, and C. Gravino, “Detecting Privacy Requirements from User Stories with NLP Transfer Learning Models,” Information and Software Technology, vol. 146, 2022.

[3] European Union. (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council (General Data Protection Regulation)*.

[4] B. Nuseibeh and S. Easterbrook, “Requirements engineering: A roadmap,” in *Proceedings of the Conference on the Future of Software Engineering (FOSE)*, Limerick, Ireland, 2000, pp. 35–46.

[5] State of California, *California Consumer Privacy Act (CCPA), Cal. Civ. Code § 1798.100–1798.199*, 2018.

[6] W. Baldwin, S. Chintakuntla, S. Parajuli, A. Pourghasemi, R. Shanz, and S. Ghanavati, “Generating Privacy Stories From Software Documentation,” arXiv preprint arXiv:2506.23014, 2025.

[7] G. B. Herwanto, G. Quirchmayr, and A. M. Tjoa, “PrivacyStory: Tool Support for Extracting Privacy Requirements from User Stories,” in Proc. IEEE 30th Int. Requirements Engineering Conf. (RE), 2022, pp. 264–265.

[8] Breaux, T. D., Vail, M. W., & Anton, A. I. (2006). *Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations*.

[9] Breaux, T. D., & Gordon, D. G. (2013). *Regulatory requirements traceability and analysis using semi-formal specifications*.

[10] Zeni, N., Kiyavitskaya, N., Mich, L., Cordy, J. R., & Mylopoulos, J. (2015). *GaiusT: supporting the extraction of rights and obligations for regulatory compliance*.

[11] A. Sleimi, N. Sannier, M. Sabetzadeh, L. Briand, M. Ceci, and J. Dann, “An Automated Framework for the Extraction of Semantic Legal Metadata from Legal Texts,” *Requirements Engineering*, 2020

[12] D. Torre, S. Abualhaija, M. Sabetzadeh, L. Briand, K. Baetens, P. Goes, and S. Forastier, “An AI-assisted Approach for Checking the Completeness of Privacy Policies Against GDPR,” *Requirements Engineering*, 2020.

[13] P. Sangaroonsilp, H. K. Dam, M. Choetkiertikul, C. Ragkhitwetsagul, and A. Ghose, “A Taxonomy for Mining and Classifying Privacy Requirements in Issue Reports,” arXiv preprint arXiv:2101.01298, 2023.

[14] F. Dalpiaz, Requirements data sets (user stories), 2018. doi:10.17632/7zbk8zsd8y.1, https://data.mendeley.com/datasets/7zbk8zsd8y/.

[15] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, MN, USA, 2019, pp. 4171–4186.

[16] V. Sanh, L. Debut, J. Chaumond and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” arXiv preprint arXiv:1910.01108, 2019.

[17] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[18] Meta AI, “Llama 3.2: Open and Efficient Foundation Models,” Meta, 2024. [Online]. Available: https://ai.meta.com/llama/

[19] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, “A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications,” arXiv preprint arXiv:2402.07927, 2024.

[20] H. He and E. A. Garcia, “Learning from imbalanced data,” IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009.

[21] G. Tsoumakas and I. Katakis, “Multi-label classification: An overview,” International Journal of Data Warehousing and Mining, vol. 3, no. 3, pp. 1–13, 2007.

[22] G. M. Kapitsaki, M. Papoutsoglou, C. Treude, and I. Theophilou, “Analyzing developer discussions on EU and US privacy legislation compliance in GitHub repositories,” arXiv preprint, 2025.

[23] J. Ramos, “Using TF-IDF to determine word relevance in document queries,” in *Proceedings of the First Instructional Conference on Machine Learning*, Piscataway, NJ, USA, 2003, pp. 133–142.

[24] A. Mishra and S. Vishwakarma, “Analysis of TF-IDF Model and its Variant for Document Retrieval,” in *Proceedings of the 2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, 2015, pp. 772–776.

[25] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[26] N. Martí and J. García, “A Core Ontology for Privacy Requirements Engineering,” arXiv preprint arXiv:1611.10097, 2016.

[27] J. R. Landis and G. G. Koch, “The measurement of observer agreement for categorical data,” *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.